SHUJU WAJUE SUANFA JIQI YINGYONG YANJIU

数据挖掘算法及其应用研究

蔡丽艳 著

电子科技大学出版社

策划编辑：罗丹
责任编辑：李述娜
封面设计：



数据挖掘算法及其应用研究

ISBN 978-7-5647-1481-9
9 787564 714819 >
定 价：29.00元

SHUJU WAJUE SUANFA JIQI YINGYONG YANJIU

数据挖掘算法及其应用研究

蔡丽艳 著

电子科技大学出版社

图书在版编目（CIP）数据

数据挖掘算法及其应用研究 / 蔡丽艳著. —成都：电子科技大学出版社，2013.2

ISBN 978-7-5647-1481-9

Ⅰ. ①数… Ⅱ. ①蔡… Ⅲ. ①数据采集—研究 Ⅳ. ①TP274

中国版本图书馆 CIP 数据核字(2013)第 018956 号

数据挖掘算法及其应用研究

蔡丽艳 著

出　　版： 电子科技大学出版社（成都市一环路东一段 159 号电子信息产业大厦　邮编：610051）
策划编辑： 罗丹
责任编辑： 李述娜
主　　页： www.uestcp.com.cn
电子邮箱： uestcp@uestcp.com.cn
发　　行： 新华书店经销
印　　刷： 成都蓉军广告印务有限责任公司
成品尺寸： 185mm×260mm　**印张** 12　**字数** 304 千字
版　　次： 2013 年 2 月第一版
印　　次： 2013 年 2 月第一次印刷
书　　号： ISBN 978-7-5647-1481-9
定　　价： 29.00 元

■ 版权所有　侵权必究 ■

◆ 本社发行部电话：028-83202463；本社邮购电话：028-83201495。
◆ 本书如有缺页、破损、装订错误，请寄回印刷厂调换。

内 容 简 介

数据挖掘是一个融合了数据库技术、人工智能、机器学习、统计学、知识工程、面向对象方法等新技术的多学科交叉的研究领域。今天，越来越多的人们投入数据挖掘技术的研究领域，各种算法的不断改进和推陈出新，再加上高性能的关系数据库技术以及数据仓库技术的成熟，使得数据挖掘技术有了飞速的发展，并使数据挖掘技术进入了实用的阶段。

本书结合大量国内外最近几年数据挖掘的最新资料和作者的研究成果，系统地介绍了数据挖掘算法以及相关的技术及其在一些领域中的应用。本书共分2个部分，第1部分介绍数据挖掘算法，包括决策树算法、贝叶斯网络算法、人工神经网络、支持向量机、关联规则、聚类分析以及一些数据挖掘的相关技术等。第2部分主要讨论数据挖掘的应用研究，包括在信用评估模型中的应用和数据挖掘技术在决策支持系统中的应用。

本书的读者可以是对数据挖掘感兴趣的计算机专业人士，也可供数据挖掘、机器智能、商业数据分析等领域的科技人员和高校师生参考。

前　言

数据收集和数据存储技术的快速发展使得各个组织机构积累了海量数据。人们也迫切需要从这些海量数据中提取有用的信息。因此，数据挖掘是在人们面对“数据丰富，但知识贫乏”的条件下应运而生的，并得以迅速发展，越来越显示出其强大的生命力。数据挖掘是致力于数据分析和理解，揭示数据内部蕴藏知识的技术。它成为未来信息技术应用的重要目标之一。经过十几年的努力，数据挖掘产生了许多新概念和方法，特别是最近几年一些基本概念和方法趋于清晰，它的研究正向着更深入的方向发展。像其他新技术的发展历程一样，数据挖掘技术也必须经过概念提出、概念接受、广泛研究和探索逐步应用和大量应用等阶段。从目前的现状看，大部分学者认为数据挖掘的研究仍然处于广泛研究和探索阶段，迫切需要在基础理论应用模式系统构架以及挖掘算法和挖掘语言等方面进行创新。

本书综合了大量国内外最近几年的数据挖掘的研究成果，以及作者几年来数据挖掘中的研究成果，介绍了数据挖掘的一些主要算法以及算法在一些领域内的应用研究。

本书从结构上分为两个部分，第1篇为数据挖掘算法篇，包括第1～第8章。第1章主要介绍了数据挖掘的概念及其产生背景、数据挖掘的起源、数据挖掘的任务、数据挖掘过程、数据挖掘解决的典型问题以及数据挖掘常用工具及开发商。从第2章开始，介绍了目前流行的主要挖掘算法。第2章为决策树算法，讨论了分类原理，决策树算法概述，决策树相关概念及定义，决策树生成过程，决策树剪枝技术以及ID3算法、C4.5算法、CART算法、SLIQ算法、SPRINT算法和PUBLIC算法的详细介绍、比较研究和应用实例。第3章为贝叶斯分类算法，主要讨论了贝叶斯分类理论、贝叶斯分类模型、贝叶斯分类特点、朴素贝叶斯分类器以及贝叶斯分类模型的改进研究，还讨论了贝叶斯网络以及贝叶斯网络的应用实例。第4章为人工神经网络算法，主要讨论了人工神经网络的基本概念及特征、人工神经网络的分类、人工神经网络的发展历史、人工神经元模型以及BP神经网络、BP神经算法的应用实例。第5章为支持向量机，主要讨论了统计学习理论、支持向量机的定义、核函数以及支持向量机的训练算法和支持向量机的应用实例。第6章为关联规则，主要讨论了关联规则相关定义及基本概念、APRIORI算法、从频繁项集到强关联规则、提高APRIORI算法的效率问题、Web挖掘和关联规则在Web挖掘中的应用实例。第7章为聚类分析，主要讨论了

聚类分析的基本概况、聚类分析涉及的数据结构与数据类型、聚类准则函数、主要聚类算法，详细探讨了K－means聚类算法以及K－means算法应用实例研究。第8章为数据挖掘相关技术，主要讨论了粗糙集技术、模糊集技术以及数据仓库技术。

第2篇为数据挖掘算法的应用篇，包括第9章和第10章。第9章主要探讨了数据挖掘在信用评估模型中的应用研究，其中主要探索了数据挖掘中的决策树算法在小额农户贷款信用评估模型中的应用。第10章主要讨论了数据挖掘在决策支持系统中的应用，并以具体案例探索了数据挖掘技术在决策支持系统中的应用研究。

本书可作为对数据挖掘感兴趣的计算机专业人士的参考书，也可作为数据挖掘、机器智能、商业数据分析等领域的科技人员和高校师生的参考用书。

作者在本书的撰写过程中，得到所在单位牡丹江师范学院工学院的领导和同事们的支持和帮助，特别是蔡静颖、张丹、杨治秋等老师提出宝贵意见，在此一并表示感谢。

本书受牡丹江师范学院科学技术研究项目“基于数据挖掘的高校就业管理决策系统的研究”（KY201102）、黑龙江省智能教育与信息工程重点实验室开放基金项目“基于RFID的物联网关键技术探讨及其在智能校园中的应用研究”（IEF2012－01）、牡丹江师范学院青年学术骨干项目“物联网模式下计算机实验室的管理与创新”（G201207）、牡丹江师范学院青年指导项目“云计算技术在农村信息化建设中的应用”（QZ201213）基金资助。

由于作者水平和时间的限制，书中一定存在一些缺点和错误，恳请读者批评指正，以便以后及时更正。

蔡丽艳

2012年12月

目　录

第 1 章　概论 …… (1)
1.1　数据挖掘概念及其产生的背景 …… (1)
1.2　数据挖掘起源 …… (2)
1.3　数据挖掘任务 …… (2)
1.4　数据挖掘过程 …… (5)
1.5　数据挖掘解决的典型问题 …… (6)
1.6　数据挖掘常用工具及开发商 …… (7)
第 2 章　决策树算法 …… (10)
2.1　分类原理 …… (10)
2.2　决策树算法概述 …… (11)
2.3　决策树相关概念及定义 …… (13)
2.4　决策树生成过程 …… (14)
2.5　决策树算法特点 …… (15)
2.6　决策树剪枝技术 …… (16)
2.7　决策树算法国内外研究现状 …… (18)
2.8　ID3 算法 …… (20)
2.9　C4.5 算法 …… (26)
2.10　CART 算法 …… (31)
2.11　SLIQ 算法 …… (32)
2.12　SPRINT 算法 …… (34)
2.13　PUBLIC 算法 …… (35)
2.14　决策树算法比较 …… (36)
第 3 章　贝叶斯分类算法 …… (37)
3.1　贝叶斯分类理论 …… (37)
3.2　贝叶斯分类模型 …… (39)
3.3　贝叶斯分类模型的改进研究 …… (43)
3.4　贝叶斯网络 …… (46)
3.5　贝叶斯网络应用实例 …… (48)
第 4 章　人工神经网络算法 …… (52)
4.1　基本概念及特征 …… (52)
4.2　人工神经网络的分类 …… (53)
4.3　人工神经网络的发展历史 …… (55)

4.4 人工神经元模型 …… (56)
4.5 BP 神经网络 …… (60)
4.6 BP 神经算法应用实例 …… (64)
第 5 章 支持向量机 …… (74)
5.1 概述 …… (74)
5.2 统计学习理论 …… (74)
5.3 支持向量机定义 …… (77)
5.4 核函数 …… (79)
5.5 支持向量机的训练算法 …… (80)
5.6 支持向量机应用实例 …… (84)
第 6 章 关联规则 …… (86)
6.1 关联规则相关定义及基本概念 …… (87)
6.2 APRIORI 算法 …… (88)
6.3 从频繁项集到强关联规则 …… (92)
6.4 提高 APRIORI 算法的效率 …… (93)
6.5 Web 挖掘 …… (98)
第 7 章 聚类分析 …… (105)
7.1 概述 …… (105)
7.2 数据结构与数据类型 …… (106)
7.3 聚类准则函数 …… (109)
7.4 主要聚类算法 …… (111)
7.5 K – means 聚类算法 …… (113)
7.6 K – means 算法应用实例 …… (118)
第 8 章 数据挖掘相关技术 …… (124)
8.1 粗糙集技术 …… (124)
8.2 模糊集技术 …… (127)
8.3 数据仓库 …… (131)
第 9 章 数据挖掘在信用评估模型中的应用 …… (141)
9.1 我国商业银行运用数据挖掘技术的必要性 …… (141)
9.2 信用的基本概念 …… (143)
9.3 信用风险的影响因素 …… (144)
9.4 风险评估的特点 …… (145)
9.5 信用评估模型的主要应用技术 …… (145)
9.6 基于数据挖掘的农户小额贷款信用评估模型研究 …… (145)
第 10 章 数据挖掘在决策支持系统中的应用 …… (158)
10.1 决策支持系统 …… (158)
10.2 决策支持系统的主要特征及功能 …… (160)
10.3 智能决策支持系统 …… (161)
10.4 基于数据挖掘的房地产决策支持系统的开发 …… (163)
参考文献 …… (178)

第1章 概 论

1.1 数据挖掘概念及其产生的背景

随着数据收集和数据存储技术的快速进步使得人们积累的数据越来越多，在这些剧增的数据背后蕴藏着许多有用而重要的信息，人们越来越迫切需要对这些数据进行更深入的分析，希望能够得出对实践有指导意义的信息。尤其在当今这个竞争世界中，吸取隐藏在这些数据背后的有用知识并利用这些知识的能力变得越来越重要。为了实现这个目的，人们发展了各种数据挖掘算法：统计分析、决策树、神经网络以及针对各种特定的实际问题的解决方法。

而传统的数据库处理技术主要是对数据进行以下操作：数据的建立、删除、查询、排序、索引等数据处理。而这些数据库技术并不能从大量的数据中发现数据背后隐藏的关系和规则，也就无法根据现有的数据预测未来的发展趋势。随着数据量的爆炸性增长，同时又缺乏发现数据背后隐藏知识的手段，因此就导致了“数据爆炸但知识贫乏”的现象，数据挖掘技术就是在这样一个背景下逐步发展起来的。

数据挖掘是一个反复迭代的过程，它从大量的数据中搜寻有价值的、能对未来有一定影响或者有一定决策指导的新信息，是人和计算机有机结合、共同作用的结果。并非所有的信息发现任务都是数据挖掘，例如，使用数据库管理系统（DBMS）查询个别记录，或者通过互联网的搜索引擎技术查找特定的 Web 页面，这些仅仅是信息搜索领域的任务。

随着数据量的日益扩大、数据库技术的快速发展以及各种算法的逐渐成熟，数据挖掘技术的研究有了长足而快速的发展。数据挖掘最早的雏形是机器学习，机器学习的基本过程就是将一些已知的并已经被成功解决的问题作为实例输入计算机中，然后机器学习就通过学习这些实例而自动生成规则，这些规则的通用性可以解决某一类问题。随着神经网络技术的出现，人们将数据处理技术的研究着眼于知识工程，知识工程的基本过程不同于机器学习，而是直接给计算机输入已被代码化的规则，然后计算机通过学习这些规则来解决某些问题。但是这种方法具有投资大、效果不理想等缺点。20 世纪 80 年代，人们在重新认识神经网络理论的前提下，又重新开始研究机器学习的方法，并将这些研究成果应用于一些大型的商业数据库中，使数据挖掘技术迈出了坚实的一步。20 世纪 80 年代末，“数据库中的知识发现”（简称 KDD）这一术语的出现使人们能够用 KDD 来描述数据挖掘的基本过程。

在进行数据挖掘的过程中，人们也逐渐认识到数据源的存储和维护的问题，于是就产

生了数据仓库技术。数据仓库可以将不同数据源的数据经过加工处理后进行存储和维护，为数据挖掘工作提供了必要的基础。当然，数据仓库并不是数据挖掘的先决条件，有很多数据挖掘是可以从直接操作数据源中挖掘信息的。数据挖掘技术经历了十多年的发展，在发展过程中离不开数理统计、机器学习、人工智能等多领域的研究成果。今天，越来越多的人投入到数据挖掘技术的研究领域，各种算法的不断改进和推陈出新，再加上高性能的关系数据库技术以及数据仓库技术的成熟，使得数据挖掘技术有了飞速的发展，并使数据挖掘技术进入了实用的阶段。

数据挖掘技术的发展过程就必然形成了数据挖掘技术具有交叉性学科的特点，主要包括了以下几门学科：数据库技术、人工智能、统计学、机器学习、信息科学、模式识别、高性能计算、可视化技术等。从商业应用的角度看，数据挖掘是一种崭新的商业信息处理技术。可以从积累了大量业务数据的商业数据库中进行数据清洗、抽取、转化、分析和一些模式化处理，从中提取能够辅助商业决策的重要知识。

数据挖掘技术是计算机行业中发展最快的领域之一，数据挖掘的强大力量之一在于它具有广泛的方法和技术，以应用于大量的问题集。数据挖掘技术可以用来支持广泛的商务智能应用，例如在顾客行为分析、营销策略决策、信用风险评估以及欺诈检测等。数据挖掘技术已经成为数据库理论和应用的一个很重要的方向，虽然数据挖掘也已经迅速扩大成为一个独立的研究领域，但是和新的数据库研究方向还是紧密相关的，例如数据仓库可看做数据挖掘的一个数据预处理过程，移动数据库、互联网数据库、海量数据库、并行数据库、空间数据库等新型数据库的挖掘方法是数据挖掘的前沿课题之一。

1.2 数据挖掘起源

数据挖掘起源于多种学科，其中最主要来源于以下两大领域：一是统计学中的抽样、估计和假设检验，二是人工智能、模式识别和机器学习的搜索算法、建模技术和学习理论。当然，随着数据挖掘迅速发展和研究的不断深入，也吸收了来自其他领域的思想，其中包括最优化、进化计算、信息论、信号处理、信息检索以及可视化。这些领域中，最需要数据库系统提供有效的存储、索引和查询处理的支持。图 1－1 描述了数据挖掘的起源。

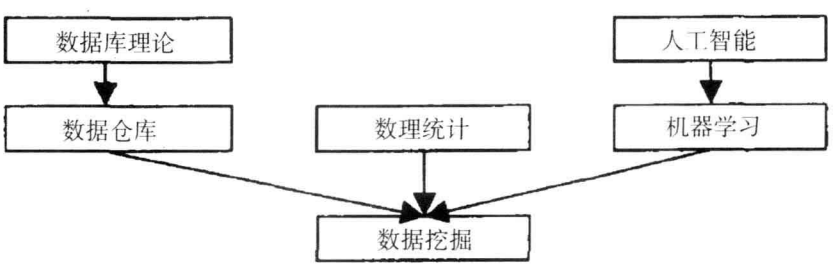


图 1－1　数据挖掘来源

1.3 数据挖掘任务

在实践中，数据挖掘的两个基本目标就是预测和描述。一方面，预测主要是使用数据

集中的一些变量或域来预测其他变量的未知的或者未来的值；另一方面，描述则是要导出概括数据中潜在的联系模式，例如相关性、未来发展趋势、聚类分析、轨迹和异常检测等方面。本质上，描述性的数据挖掘任务常常是那种探查性质的，并且常常需要后处理技术去验证和解释导出的结果模式。数据挖掘的基本任务总结如下：

（1）分类和预测

分类和预测是数据分析形式，主要用于提取描述重要数据类的模型或预测未来的数据趋势。可以用来预测数据对象的类标记，如果被测试的值是数值型的数据时，就称之为预测。分类是找到数据库中一组对象的相同属性，根据分类模型将它们分为不同类别的过程。它是数据挖掘中应用得最多的任务。设示例数据库中的记录由一组特征描述，每个记录有一个类别标记，分类则是通过分析示例数据库中的数据，为每个类别做出准确的描述，建立分类模型，然后用这个分类模型对其他未知类别的记录进行分类。分类方法有很多，包括决策树方法、贝叶斯方法、神经网络、遗传算法和粗糙集方法等。分类也能进行预测，但分类一般用于离散型数值。回归预测用于连续型数值。近年来发展起来的神经网络方法，既可用于连续数值预测，也可以用于离散数值预测。

（2）关联分析

关联分析由 Agrawal、Imielinski、Swami 在 1993 年提出，它是数据挖掘最先研究的问题之一，也是数据挖掘的一个主要研究方向。数据库中的数据一般都存在着关联关系，也就是说，两个或多个数据项的取值之间重复出现且概率很高时，它就存在某种关联，可以建立起这些数据项的关联规则。例如，买面包的顾客有 90% 的人还买牛奶，这是一条关联规则。规则通常用如下形式：$X \rightarrow Y$ 表示，可以解释为满足 X 的数据库元组也很可能会满足 Y。还存在时序关联，例如：若 AT&T 股票连续上涨两天且 DEC 股票不下跌，则第三天 IBM 股票上涨的可能性为 75%。它在简单关联中增加了时间属性。

关联分析的主要目的是找出数据库中隐藏的关联网，描述一组数据项目的密切或关系。用户并不知道数据库中的数据的关联是否存在精确的关联函数，即使知道也是不确定的，因此关联分析生成的规则带有置信度，置信度级别度量了关联规则的强度。关联规则的一个典型例子就是购物篮分析（Marketing Bsaket Analysis），通过挖掘数据派生关联规则，可以了解客户的行为，进而可以对商业促销提供决策信息，实现在商品销售方面做各种促销活动。在大型数据库中，这种关联规则是很多的，需要进行筛选，一般用“支持度”和“可信度”两个阈值来淘汰那些无用的关联规则。

（3）聚类

遇到要分析的数据缺乏描述性信息时，或者是无法组织成任何分类模式时，就可以采用聚类分析。数据库中的数据可以划分为一系列有意义的子集，即类。聚类就是把一组对象按照属性值划分成一系列有意义的子集的过程。它是把一组未被标记的数据作为输入，根据数据自身的距离或相似度进行划分。它所采用的划分原则是保持最大的组内相似性和最小的组间相似性，即不同的聚类中的数据尽可能地不同，同一聚类中的数据尽可能相似。所以，聚类有助于建立集合的细分。聚类的方法有很多，包括层次聚类法、划分聚类法、密度聚类法、模糊聚类法和网络法等。通常，对于同一个数据集合，采用不同的聚类方法可能会得到不同的结果。

在机器学习中，聚类分析通常称为“无监督学习”。因为和分类学习相比，分类学习的样本或数据对象有类别标识，而聚类学习的样本则没有，需要聚类学习算法自动确定。

也可称为“概念聚类”，因为其中的距离衡量不再是基于几何距离，而是基于一组对象的描述。执行概念聚类时，需要先定义对象间相似性的量度，然后根据它确定类。确定的类为一组对象的集合，类内相似性很高而类之间相似性很低。

神经网络中的一类无监督自学习方法即自组织神经网络方法可进行聚类分析，如 Kohonen 自组织特征映射方法和竞争学习网络。

聚类分析和分类分析在实际应用中可以结合使用，例如在最初的分析中，分析人员根据以往的经验先利用聚类分析将要分析的数据进行标定，划分类别，然后用分类分析方法分析该数据集合，挖掘出每个类别的分类规则；接着用这些分类规则重新对这个集合（抛弃原来的划分结果）进行划分，以获得更好的分类结果。这样，分析人员可以循环使用这两种分析方法，直至得到满意的结果。

（4）偏差分析

数据库中的数据存在很多异常情况，从数据分析中发现这些异常情况也是很重要的，以引起人们对它更多的注意。偏差包括很多有用的知识，如分类中的反常实例，模式的例外，观察结果对模型预测的偏差量值随时间的变化。偏差检测的基本方法是寻找观察结果与参照之间的差别。观察常常是某一个域的值或多个域值的汇总。参照是给定模型的预测、外界提供的标准或另一个观察。

（5）概念描述

概念描述是描述性数据挖掘中最简单的类型，即对某类对象的内涵进行描述，并概括这类对象的有关特征。通常数据库中存放大量的细节数据，通过概念描述，用户可以以一种简洁的描述形式观察汇总的数据集。概念描述分为特征性描述和区别性描述，前者描述某类对象的共同特征，后者描述不同类对象之间的区别。

可视化系统则是为使数据挖掘能以图形或图像的方式在屏幕上显示出来，且能交互处理，可以很清楚地发现隐含的和有用的知识。可视化技术可分为两类：表示空间数据的可视化技术和表示非空间数据的信息可视化。可视化数据挖掘可以分为数据可视化、数据挖掘结果可视化、数据挖掘过程可视化和交互式数据可视化挖掘。空间数据挖掘则是基于地理信息系统的数据挖掘技术。地理信息系统（GIS）的应用现已扩展到航天、电信、电力、交通运输、商业、市政基础设施管理、公共卫生及安全、油气等其他矿产资源的勘测等诸多领域。在这些领域中的数据挖掘技术可用于地图、预处理后的遥感数据、医学图像数据和 VLSI 芯片设计空间数据库中非显式的知识、空间关系和其他有意义的模式的提取。空间数据挖掘方法目前主要有空间数据分类、空间数据关联分析和空间趋势分析等。

分布式数据挖掘是基于分布式数据库的，利用分布式算法从分布式数据库中挖掘知识的技术。分布式数据挖掘技术主要用于对水平方式分布或垂直方式分布的数据库系统数据的挖掘。水平分布式数据挖掘算法只需要首先完成各个站点的局部数据分析，构造局部数据模型，最后组合不同数据站点上的局部数据模型，获得全局数据模型即可。垂直式分布的数据库系统则需要采用汇集型数据挖掘方法来实现。分布式数据挖掘将更有利于对分布式数据库数据资源的利用。商业领域中数据挖掘技术较多且较成熟的是统计分析类的数据挖掘技术，而从商业用角度看，数据挖掘的常用技术一般是知识发现类的技术，尤其是分类与预测。

1.4 数据挖掘过程

数据挖掘中被研究的业务对象是整个过程的基础，它驱动了整个数据挖掘过程，也是检验最后结果和指引分析人员完成数据挖掘的依据和顾问。数据挖掘过程并不是自动的，其中绝大多数工作需要人工完成。

由数据挖掘的发展过程可以得知，数据挖掘只是发现过程的一个基本步骤。过程中的主要步骤内容如下：

1. 确定业务对象

明确而清晰地定义业务问题，确定数据挖掘的目的。数据挖掘的最后结果是不可预测的，但是要探索的问题应具预见性；而数据挖掘则带有盲目性，是不会成功的。

2. 数据准备

（1）数据选择

数据挖掘的工作主要开始在大型数据库或数据仓库目标中提取数据挖掘所需要的目标数据集。

（2）数据预处理

主要是要对数据进行再加工，包括分析数据的完整性及一致性，去除数据噪声，填补缺少的属性值，删除无效数据等。

（3）数据转换

将数据转换成主要是针对挖掘算法的一个分析模型，即将数据转换或统一成适合于挖掘的形式。是否能够建立一个真正适合挖掘算法的分析模型是数据挖掘成功的关键。

3. 数据挖掘

将转换的数据进行挖掘，根据数据功能的类型及数据的特点选择合适的算法，在经过数据处理和转换的数据集上进行数据挖掘。

4. 模式评估

根据一定的评估标准从挖掘的结果集中筛选出有意义的模式知识。

5. 知识表示

其主要作用就是利用可视化和知识表达技术，转换成能够被用户理解的知识，最终向用户提供数据挖掘的知识。

综上所述，数据挖掘的全过程描述如图 1－2 所示。

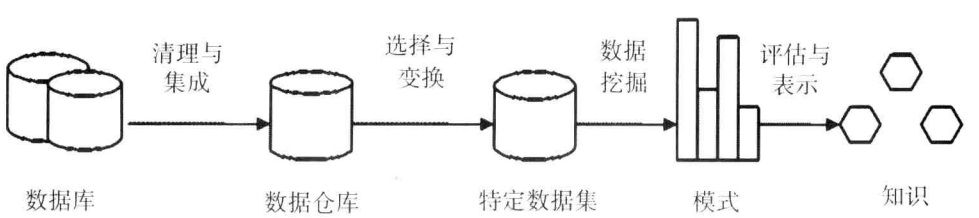


图 1－2　数据挖掘过程示意图

1.5 数据挖掘解决的典型问题

需要强调的是，数据挖掘技术从一开始就是面向应用的。数据挖掘技术的应用领域十分广泛，从市场营销、政府管理决策、科学研究到工业决策支持等各个领域。

（1）市场营销

市场营销是数据挖掘技术应用最早也是最重要的领域，它主要包括零售业的市场营销、信用卡业的市场营销、电信业的市场营销等。如，数据挖掘技术可以通过对顾客购货篮的分析，协助商家发现顾客的消费习惯，预测顾客的购买模式，划分顾客群体，发现商品销售量之间的关系，优化货架布置，研究最佳的促销活动时间和促销商品组合等商业活动。

（2）金融领域

金融事务需要搜集和处理大量数据，对这些数据进行分析，发现其数据模式及特征，然后可能发现某个客户、消费群体或组织的金融和商业兴趣，并可观察金融商场的变化趋势。利用数据挖掘技术可以帮助银行产品开发部门描述客户以往的需求趋势，并预测未来。在金融领域中的应用主要包括了金融市场的预测和分析、客户分类、银行担保以及信用风险评估等，其中信用风险评估的研究是本文主要研究目标和内容。

（3）保险业的风险评估

由于金融投资风险很大，在进行投资决策时，要对各种投资方向的相关数据进行分析，选择最佳的投资方向。风险评估对保险公司的正常工作起着重要的指导作用，保费和保单的设计都需要详细的风险分析。通常，保费是根据一些主要因素人为地直观分析确定，而利用数据挖掘分析风险投资可以得到一些有用的规则，以确定适当的赔偿金额。

（4）股票

股票市场是一个典型的具备海量数据的行业，中国上海的证券交易所每天产生超过1GB的数据，那么投资者可以利用数据挖掘技术预测股票价格的变化，分析各只股票之间的关系，预测股票价格的升降及决定交易的最佳时间等。

（5）工程

数据挖掘技术在工程技术领域中应用较为广泛，并已取得了一些重要成果。如法国的Brezillon等人利用数据挖掘技术对地铁发生事故的时间、地点以及驾驶员本身的能力等多方面的因素进行挖掘，成功地应用于地铁交通调度智能系统。另外，数据挖掘技术还应用于工程中的图像识别、制造业、机器人导航等。其中，E. Swere 和 D. J. Mulvaney 将数据挖掘技术应用于移动机器人导航，并取得了一定的成功。

（6）Web 挖掘

Web 挖掘就是从 WWW 的资源和行为中抽取感兴趣的、有用的模式和隐含的信息。如将数据挖掘技术应用在相关主页的 Web 访问日志上以发现顾客的偏好与行为，也可根据因特网的访问情况分析电子商务的可行性，并改进 Web 网站的组织结构。另外，还可以在搜索引擎上对文档进行自动分类及利用数据挖掘技术设计电子邮件过滤系统。

1.6 数据挖掘常用工具及开发商

目前国际上广泛应用的数据挖掘工具有很多，如SAS公司的Enterprise Miner，SPSS公司的Clementine，Waikato大学开发的Weka平台，SQL Sever的数据挖掘模块，IBM公司的DB Miner等。SAS（Statistical Analysis System）是由美国North Carolina州立大学1966年开发的统计分析软件。SAS是一个模块化、集成化的大型应用软件系统。它由数十个专用模块构成，功能包括数据访问、数据储存及管理、应用开发、图形处理、数据分析、报告编制、运筹学方法、计量经济学与预测等等。SAS系统基本上可以分为四大部分：SAS数据库部分；SAS分析核心；SAS开发呈现工具；SAS对分布处理模式的支持及其数据仓库设计。SAS系统主要完成以数据为中心的四大任务：数据访问、数据管理、数据呈现、数据分析。SAS持续良好的统计分析功能，得到了业界广泛好评，这为它在国际专业统计分析软件领域获得头把交椅奠定了基础。

1. SAS Enterprise miner模块介绍

SAS/Enterpreise Miner是在数据挖掘市场上令人敬畏的竞争者。它支持SAS统计模块，使之具有杰出的力量和影响，它还通过大量数据挖掘算法增强了那些模块。SAS使用它的SEMMA方法学以提供一个能支持包括关联、聚类、决策树、神经元网络和统计回归在内的广阔范围的模型数据挖掘工具。SAS Enterprise Miner设计被初学者和有经验的用户使用。它的GUI界面是数据流驱动的，且它易于理解和使用。它允许一个分析者通过构造一个使用链接连接数据结点和处理结点的可视数据流图建造一个模型。另外，此界面允许把处理结点直接插入数据流中。由于支持多种模型，所以Enterprise Miner允许用户比较（评估）不同模型并利用评估结点选择最适合的。另外，Enterprise Miner提供了一个能产生被任何SAS应用程序所访问的评分模型的评分结点。

Enterprise Miner支持市场划分分析、分类、预测模型、顾客分析、计量经济时序的统计分析范围、运作研究和其他许多方面。Enterprise Miner支持两种评估模型的方法：通过模型管理器或通过评估结点。模型管理器是从特定的模型分析输出结果的好工具。评估结点在评估模型的概况和健壮性方面很有用。两者都支持提升表、利润表、ROI和别的指示器。

2. SAS Enterprise miner模块的SEMMA分析方法介绍

SEMMA是SAS公司提出的一种数据挖掘方法论。它将数据挖掘过程划分为以下几个主要步骤：数据抽样（Sample），数据特征探索和分析（Explore），数据调整与预处理（Modify），算法的选择、模型的建立或知识的发现（Model），模型或知识的综合解释与评价（Assess）。

IBM公司的Intelligent Miner系统具有典型的数据集自动生成、关联发现、序列规律发现、概念性分类以及可视化显示等功能，在数据挖掘的过程中可以实现数据选取、数据转换、数据挖掘以及结果显示等一系列操作的自动化，是一款比较优秀的数据挖掘工具。

RightPonit公司开发的Data Cruncher是一种基于客户/服务器模式的数据挖掘引擎。这

个工具能够分析数据仓库中的海量数据，并且能够与许多主流的关系数据库和数据挖掘辅助工具进行直接的连接，进而能够辅助决策者建立面向营销的数据挖掘研究的模型。

Angoss Software Corporation（加拿大）的主要产品是 Knowledge Seeker 和 Knowledge Studio。它采用决策树技术，可以根据各种用户给定的关系自动生成决策树，并用图形显示树形结构，然后用多种决策树算法帮助用户进行决策过程。

Attar Software，Inc（英国）的主要产品是 XportRule Miner。它提供了知识提取功能，有顾客终端和服务器之间的远程通信协调办法，可以优化数据系统；支持完整的数据挖掘过程；可以进行分类、聚类、预测、时间序列分析等，采用的技术有关联规则、决策树、神经网络等，可以接纳多种数据源。

SPSS，Inc（美国）在数据挖掘方面的主要产品是 SPSS Clementine。它以统计型的数据挖掘软件见长，是主要统计软件的供应商之一。其特色是它具有面向对象扩展的模块接口，该接口使用户算法和工具可以加到 Clementine，最早是由 ISL 公司开发的，现在已被 SPSS 公司收购。Clementine 含有规则归纳算法、关联规则、分类、神经网络算法和可视化工具。在规则归纳算法中，一旦被训练过，算法就创建一个决策树，用来代表规则，然后，定位最顶层的重要变量，利用这些变量产生一个神经网络。该软件的神经网络可以解决聚类问题。并且，它为终端用户和开发者提供了一个集成化的数据挖掘开发环境，用户可以选择不同的网络结构，例如层数及神经元个数以及多种训练方法。在关联规则中，它把一系列属性与一个特定的结论相关联，各个属性之间也有关联。Clementine 可用于顾客分析、时序预测、信用卡欺诈检测等领域。它是一个强大的工具，并具有可伸缩性、可预测性等优点。

Insightful Miner（I－Miner）是由美国 Insightful 公司开发的具有高度可扩展性的数据分析和数据挖掘软件。它既适用于数据挖掘初学者，又适用于有经验的数据分析专家。其主要特点是：I－Miner 可视化编程界面减少了数据挖掘的复杂性，使得广大的非高级用户也能使用复杂的预测模型建模；用户界面基于独特的管道式架构，确保高度可扩展性；所有的 I－Miner 组件，如数据获取、探索性分析、数据操纵、数据清洗、统计模型、机器学习、模型评估和发布都充分利用独特的管道式架构保证在数据规模和数据处理速度方面的可扩展性；与其他挖掘工具不同的是，所有 I－Miner 组件既能运行于驻留内存中的数据集，又能运行于驻留于内存外的数据集；I－Miner 与数据分析语言——S 语言的集成使其成为数据挖掘市面上算法最丰富和最具有可扩展性的工具之一；I－Miner 在数据挖掘工具市场上设立了新的性价比标准：低成本初始费用、提供桌面版、永久的 License 无须每年缴纳年费。Insightful Miner 目前在金融、生物科技、政府机构等企事业单位应用非常广泛。

除此以外，国外还有其他一些比较成熟的数据挖掘系统，包括：Oracle 公司从 Thinking Machines 公司取得的 Darwin；Unica 公司开发的 Affinium Model；加拿大 Simon Fraser 大学开发的 DB Miner；SGI 公司和美国 Standford 大学联合开发的 Minset；HNC 公司开发的用于信用卡诈骗分析的 Database Mining Workstation；IBM 公司 Almaden 研究中心开发的 Quest、NeoVista 开发的 Decision Series；以及 KEFIR 系统、SKICAT 系统等。这些数据挖掘系统的特点各不相同，有的适合大规模企业的长期数据挖掘项目，有的针对特殊行业的数据应用，还有的用于实验室科研机构的数据挖掘理论研究等。这些数据挖掘系统面向应

用，并且在应用中得到了广泛的推广和使用，反过来也大大推动了数据挖掘理论的研究与发展。

虽然数据挖掘工具的出现使得人们不必再掌握艰深的统计分析技术，但是技术人员首先要有良好的数据挖掘概念，而且要懂得基本的商业和行业规则和知识，他们所选用的技术和优化方法会对建立的模型的准确度和生成速度产生很大影响。因此，研究人员仍需要知道所选用的数据挖掘工具是如何工作的，所采用的算法原理是什么。

第 2 章　决策树算法

2.1　分类原理

决策树和决策规则是解决实际应用中分类问题的主要数据挖掘方法。分类在数据挖掘中是一项非常重要的任务，目前在应用领域中也是最广泛的方法。它是以实例为基础的归纳学习算法，通常采用形成分类器和预测模型，着眼于从一组无次序、无规则的事例中推理出决策树表示形成的分类规则。

要想更好地理解决策树分类算法，就要了解分类的基本原理。一般来说，分类是把数据项映射到其中一个事先定义的类中的这样一个学习函数的过程。首先，由一组输入的属性值向量和相应的类，用基于归纳学习算法得出分类。学习的目标是构建一个分类模型，通常也成为分类器。这个分类器能够根据有效的属性输入值来预测一些样本的类。也就是说，分类就是把某个不连续的标识值分配给一个未标识的记录的过程。例如：一个简单的分类可以把银行信贷客户的信贷记录分成具体的两组：超过 1 个月未按时偿还贷款客户和按期偿还贷款客户。现今由于数据量的爆炸式增长，分类过程需要自动化，研究人员致力于运用不同的分类方法。

数据分类过程主要包含两个步骤：

第一步，建立一个描述已知数据集类别或概念模型。如图 2 - 1 所示，该模型是通过数据库中各数据行的内容分析而获得的。每一数据行都可以认为是属于一个确定的数据类别，其类别值是由一个属性描述（被称为类别标记属性）。分类学习方法所使用的数据集称为训练样本集合，因此分类学习又可称为监督学习（Learning by Example）。它是在已知训练样本类别的情况下，通过学习建立相应模型；而无教师监督学习则是训练样本的类别与类别个数均未知的情况下进行的。

通常分类学习所获得的模型可以表示为分类规则形式、决策树形式或数学公式形式。例如：给定一个顾客信用信息数据库，通过学习所获得的分类规则可用于识别顾客是否具有良好的信用等级或一般的信用等级。分类规则也可用于对（今后）未知（所属类别）的数据进行识别判断，同时也可以帮助用户更好地了解数据库中的内容。

姓名	年龄	收入	信用等级
王宇	<30	低	一般
赵勇	30~40	低	良好
李明	<30	高	良好
……	……	……	……

图 2-1　数据分类第一步：学习建模

第二步，利用所获得的模型进行分类操作。如图 2-2 所示，首先对模型分类准确率进行估计。保留（Holdout）方法是一种简单的估计方法。它利用一组带有类别的样本进行分类测试（测试样本随机获得且与训练样本相互独立）。对于一个给定数据集所构造出模型的准确性可以通过由该模型所正确分类的（测试）数据样本个数所占总测试样本比例得到。

姓名	年龄	收入	信用等级
赵丹	>30	高	一般
宋丽	<30	低	一般
蔡春华	30~40	高	良好
……	……	……	……

图 2-2　数据分类第二步：分类测试

对于每一个测试样本，其已知的类别与学习所获模型的预测类别进行比较。若模型的准确率是通过对学习数据集的测试所获得的，这样由于学习模型倾向于过分逼近训练数据，从而造成对模型测试准确率的估计过于乐观。因此需要使用一个测试数据集来对学习所获得的准确率进行测试工作。如果一个学习所获得的模型的准确率经测试被认为是可以接受的，那么就可以使用这一模型对未来数据行或对象（其类别未知）进行分类。

综上所述，分类知识是反映同类事物共同性质的特征知识和不同事物之间的差异型特征知识。分类的目的是学会通过一个分类函数或分类模型把数据库中的数据项映射到给定类别中的某一个。最为典型的分类方法是基于决策树的分类算法。

2.2　决策树算法概述

自 20 世纪 60 年代以来，决策树在分类、预测、规则提取等领域有着广泛应用，特别是在 Quinlan JR 于 1986 年提出 ID3 算法以后，决策树方法在机器学习、知识发现领域得到进一步应用及巨大的发展，在人工智能领域有着相当重要的理论意义与实用价值。

决策树（Decision Tree）是一个类似流程图的树型结构，树中包含三种节点类型：根节点、内部节点以及叶节点。其中树中的每个内部节点都代表着对一个属性的测试，用于区分具有不同特性的记录。叶节点则代表某个类或者是类的分布，最上面的节点就是根节点。决策树构造的输入是一组带有类别标记的数据，往往构造的结果是一棵二叉树或多叉

树。

决策树分类方法首先要利用训练集建立决策树模型，然后根据这个决策树模型对输入数据进行分类。其中，关键问题在于决策树的构建过程，这一过程包括建树和剪枝两个步骤：第一个步骤为建树阶段；第二个步骤为剪枝阶段。决策树的建树算法是通过递归过程，最终得到一棵决策树，而剪枝则是为了降低噪声数据对分类正确率的影响。

决策树是一种有指导的学习方法，该方法通常是先根据训练子集构建一个决策树，剩下的训练集实例用于已构建的决策树的评估检验。如果决策树能够正确地对测试的实例进行分类，则构造决策树过程结束。如果某实例分类有误，就将该实例添加到所选的训练子集中，并且继续构造一棵新的树，直到创建了一个能正确分类所有未选的训练实例的决策树为止。

决策树是一个能够自动对数据进行分类的树型结构，可以直接地转换为决策规则，或者说是分类规则，因此它常被看成一棵树的预测模型。决策树也可以解释成一种特殊形式的规则集，并且规则集的主要特征是层次组织关系。

决策树生成基本算法如下所述：

Generate_decision_tree//根据给定数据集产生一个决策树

输入：训练样本，各属性均取离散数值，可供归纳的候选属性集为 attribute_list。

输出：决策树。

处理流程：

（1）创建一个结点 N；

（2）若该结点中的所有样本均为同一类别 D，则//开始根结点对应所有的训练样本

（3）返回 N 作为一个叶结点并标志为类别 D；

（4）若 attribute_list 为空，则

（5）返回 N 作为一个叶结点并标记为该结点所含样本中类别个数最多的类别；

（6）从 attribute_list 选择一个信息增益最大的属性 test_attribute；

（7）并将结点 N 标记为 test_attribute；

（8）对于 test_attribute 中的每一个已知取值 a_i，准备划分结点 N 所包含的样本集；

（9）根据 test_attribute = a_i 条件，从结点 N 产生相应的一个分支，以表示该测试条件；

（10）设 S_i 为 test_attribute = a_i 条件所获得的样本集合；

（11）若 S_i 为空，则将相应叶结点标记为该结点所含样本中类别个数最多的类别；

（12）否则将相应叶结点标志为 Generate_decision_tree（S_i，attribute_list-test_attribute）

返回值；

算法递归操作的终止条件是：

（1）给定结点的所有样本属于同一类（步骤 2 和 3）。

（2）没有剩余属性可以用来进一步划分样本（步骤 4）。

（3）没有样本满足 test_attribute = a（步骤 11）。

2.3 决策树相关概念及定义

（1）训练集（Training Set）

用来构造分类模型的数据集称为训练数据集或训练样本集，简称训练集。训练集由一条条数据记录组成，每一条记录（也称样本）是由有关字段值组成的一个特征向量，这些字段被称作属性。另外，训练集的每条记录还有一个特定的类标签与之对应，类标签也就是训练集的类别标记。一个具体的训练样本的形式可以表示为：$(X_1, X_2, \cdots, X_n: Y)$，其中 X_n 表示字段值，Y 表示类标签属性值，即类别。

一般情况下，训练集是根据实际需要由实际的历史数据生成的、有一定综合程度的、用于数据分析处理的数据集。训练集是决策树算法的基础，训练集中的对象被输入到决策树算法中，通过学习生成决策树。

（2）测试集（Test Set）

用来评价模型的数据集称为测试数据集或者测试样本集，简称为测试集（Test Set）。测试集一般选取与训练集不同的数据集，但它也是由已知类别的对象组成的数据集，用来测试决策树的性能。生成决策树之后，把测试集中的对象逐个输入到生成的决策树中，通过决策树判定该对象所属的类别，然后把实际类别与判定的类别进行比较，通过对比较结果的统计，可以得到决策树的分类正确率。

（3）决策树（Decision Tree，DT）

决策树是决策树分类算法的最终输出结果。决策树是一种类似流程图的树状结构，是一棵有向、无环树。根据层次的不同，决策树中的节点可分为根节点、内部节点和叶节点三类。树中的每一个节点代表数据集中的一个属性，树的最顶层为根节点，其中每个内部节点表示在一个属性上的测试，每个分枝代表一个测试输出，每个叶节点代表类。下面给出了决策树的具体定义：

定义 2.1：给定一个数据库 $D = \{t_1, \cdots, t_n\}$，其中 $t_1 = \langle t_{i1}, \cdots, t_{ih} \rangle$，数据库模式包含下列属性$(A_1, A_2, \cdots, A_h)$。同时给定类别集合 $C = \{C_1, C_2, \cdots, C_m\}$。对于数据库 D，决策树是指具有下列性质的树：

每个内部节点都被标记为一个属性 A；

每个弧都被标记一个谓词，这个谓词可应用于相应父节点的属性；

每个叶节点都被标记一个类 C_1。

如图 2-3 所示，这是一个决策树模型示意图。这棵决策树对用户的信用风险进行分类。预测出将要申请银行贷款的人的信用等级。每个内部节点（方形框）代表对某个属性的一次检测。每片叶子节点（椭圆形）代表一个类。输入一个新的被决策的记录，就可以预测该记录输入哪个类。

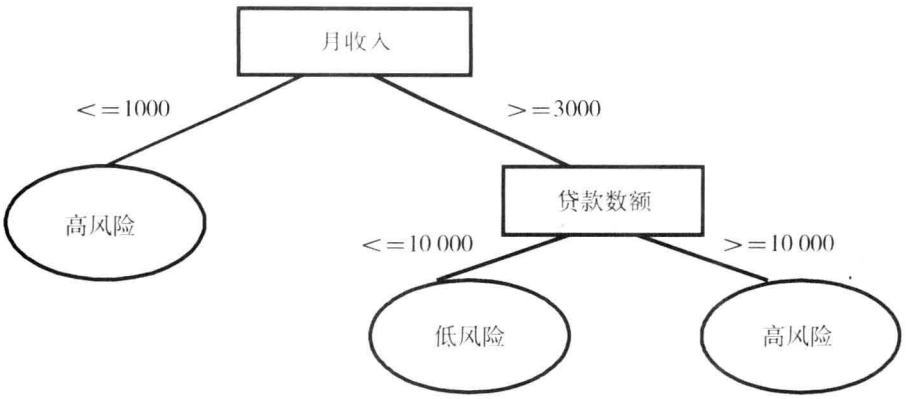


图 2-3　决策树示意描述

2.4　决策树生成过程

对一个分类问题或规则学习问题，决策树的生成是一个从上至下、分而治之的过程，本质是贪心算法。从根结点开始，对每个非叶结点，找出其对应样本集中的一个属性（称为测试属性）对样本集进行测试，根据不同的测试结果将训练样本集划分成若干个子样本集，每个子样本集构成一个新叶结点，对新叶结点再重复上述划分过程，这样不断循环，直至达到特定的终止条件。其中，测试属性的选择和如何划分样本集是构建决策树的关键环节。不同的决策树算法在此使用的技术不尽相同。

决策树采用自顶向下的递归方式，在决策树的内部结点进行属性值的比较，并根据不同的属性值判断从该结点向下的分枝，在决策树的叶结点的一条路径就对应着一条合取规则，整棵决策树就对应着一组析取表达式规则。基于决策树的学习算法的一个最大优点就是它在学习过程中不需要使用者了解很多的背景知识（这也同时是它最大的缺点），只要训练例子能够用属性—结论的方式表达出来，就能使用该算法来学习。

决策树构造过程通常有以下两个步骤：

第一步，根据给定的训练集，找到合适的映射函数 $H:F(X)\rightarrow C$ 的表示模型。即建立决策树模型。这一过程实际上是一个从数据中获取知识，进行机器学习的过程。

第二步，利用第一步训练完成的决策树模型预测数据的类别，或利用该决策树模型，对数据集中的每一类数据进行描述，形成分类规则。

其中，决策树的构建过程是关键，这一过程包括建树和剪枝两个步骤。第一个步骤为建树阶段。对训练集数据递归使用决策树算法建立决策树，直到每个叶子节点属于同一个类为止。第二个步骤为剪枝阶段。剪枝是对树结构进行修剪、删除多余分支的过程，以提高今后分类识别的速度和分类识别新数据的能力。其实质是消除训练集中的异常和噪声。

例如图 2-4 描述了决策树构造过程。

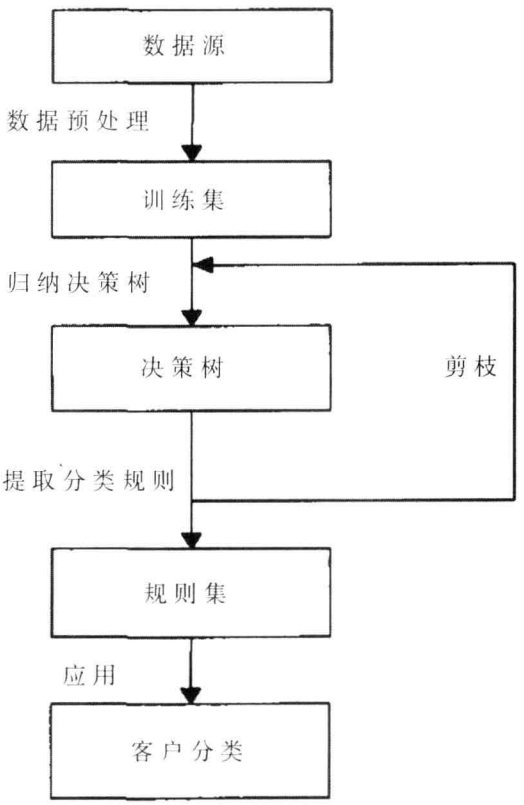


图 2-4　决策树构造过程示意图

2.5　决策树算法特点

决策树技术是应用最广泛的分类技术，下面对其特点总结如下。

决策树算法的优点：

（1）决策树模型的构造不需使用者了解很多背景知识，只要训练例子能够用属性—值结论式的方式表达出来，就能使用该算法来学习。

（2）决策树方法具有较高的分类精确度，它是指在数据库的各个对象中找出共同特性，并按照分类模型把它们进行分类。

（3）决策树方法结构简单，每个叶节点对应一条分类规则，便于人们理解。

（4）决策树可以清晰地显示出属性的重要程度。从直观上看，决策树节点距离根节点越近，该节点所代表的属性就越重要，相反节点所代表的属性对此次分类的作用就越小。

（5）决策树对于大型数据库具有很好的扩展性。由于决策树的规模独立于数据库规模，所以决策树对于大型数据库具有很好的扩展性。

决策树除了具有以上优点之外，也存在着明显的缺点。其缺点主要体现：

（1）决策树算法不易处理连续数据。数据的属性必须被划分为不同的类别才能处理。

（2）决策树处理缺失数据困难。因为决策树处理缺失数据是不能产生分枝的。

（3）决策树过程忽略了数据库中属性之间的相关性。

2.6 决策树剪枝技术

决策树的创建过程中，由于数据的噪声和孤立点等问题，许多分支反映的是训练样本集中的异常。由于训练样本集中的噪声、错误项、干扰属性等因素影响，使得根据训练集生成的决策树包含了一些错误的信息，致使分类的准确率降低，而且通常据此生成的决策树规模较大，这就是过度拟合问题。

剪枝技术的出现很好地改善了产生过度拟合的决策树的性能，剪枝方法通常分为前剪枝和后剪枝。前剪枝是通过提前停止树的构造而对决策树进行剪枝的，一旦树的构造停止，内部节点就成为叶节点。后剪枝方法是在“完全生长”的决策树上剪去不适当的分支，通过删除节点的分支，剪掉树节点，从而简化复杂而庞大的决策树。其中最典型的后剪枝算法就是代价复杂性剪枝算法，通过树中的每个非叶节点计算该节点上的子树被剪枝后可能出现的期望错误率，然后把每个分支的错误率和每个分支观察的权重进行评估，最终计算得出该节点剪枝的期望错误率。那么在对该节点进行剪枝的过程中就要考虑剪去该节点导致的期望错误率是否较高，如果较高就保留这个子树，否则就剪去该子树。这个过程需要逐次进行，需要使用一个独立的测试样本集对每棵树的测试准确率进行评估，最终得到具有最小期望错误率的决策树。

2.6.1 前剪枝（pre－pruning）

前剪枝就是在完全正确分类训练集之前，较早地停止树的生长。具体在什么时候停止决策树的生长有多种不同的方法：

（1）一种最为简单的方法就是在决策树到达一定高度的情况下就停止树的生长，这种停止标准在一定情况下能取得比较好的效果；

（2）到达此结点的实例具有相同的特征向量，而不必一定属于同一类，也可停止生长。这种情况可以处理数据中的数据冲突问题；

（3）到达此结点的实例个数小于某一个闭值时也可停止树的生长，其不足之处是不能处理那些数量较小的特殊情况实例；

（4）更为普遍的做法是计算每次扩张对系统性能的增益，如果这个增益值小于某个阈值则不进行扩展。如果在最好情况下的扩展增益都小于阈值，则即使有些叶结点的实例集不属于同一类，算法也停止。

前剪枝有一个缺点，即视野效果问题。也就是说在相同的标准下，也许当前的扩展不能满足要求，但是更进一步的扩展能够满足要求。这将使得算法过早地停止决策树的构造。因此前剪枝在决策树生成时可能会丧失一些有用的结论，而这些结论往往在决策树完全建成以后才会被发现，而且，确定何时终止决策树生长是个问题，目前使用较多的是后剪枝方法。

2.6.2 后剪枝（post－pruning）

后剪枝是人们普遍关注的树简化方法，后剪枝算法输入一个未修剪的树 T，输出修剪了 T 中一个或多个子树后获得的树 T'。算法并非搜索每个可能的 T'，而借助启发式搜索。

修剪将子树转化为叶，进而用叶节点替代内部节点。与前剪枝不同是，后剪枝没有使用一个消除细节的函数，而是直接采用默认的同质暂停规则。当树相对训练集冗余时（即噪声参与了建模），修剪将非常有效地提高精确度。例如，有训练集 m，假设包含训练集 n 的叶节点类标签为多数满足 $n' \leqslant n$，则替代错误率为 $(n - n')/m$，低层叶节点对替代精确度的影响最小，因而被最先修剪。后剪枝方法借助多种评价函数，用于确定修剪一个节点是削弱还是增强了事例集的精确度。恰当的修剪可以提高分类的精确度。尤其是当训练集噪声级别比较高时，修剪相当有效。

后剪枝技术允许决策树在充分生长的基础上，再根据一定的规则，剪去决策树中的那些不具有一般代表性的叶结点或分支。后剪枝是一个边修剪边检验的过程，一般规则是：在决策树不断剪枝的过程中，利用训练样本集或检验样本集数据，检验决策子树对目标变量的预测精度，并计算出相应的错误串。如果存在某个叶子剪去后能使得在测试集上的准确度或其他测度不降低（不会变得更坏），则剪去该叶子。

后剪枝算法有自上而下和自下而上的两种剪枝策略。自下而上的算法首先从最底层的内节点开始剪枝，剪去满足一定条件的内节点，在生成的新决策树上递归调用这个算法，直到没有可以剪枝的节点位置。自上而下的算法则是从根节点开始向下逐个考虑节点的剪枝问题，只要节点满足剪枝的条件就可以进行剪枝。

有些后期修剪方法将训练集分为两个子集。一个用于生成树，另一个用于后期修剪。不妨称之为生成集和修剪集。生成集常用于生成一个树，然后，按修剪变量生成树集 S，修剪集将从 S 中选择一个最佳的树。例外情况是除错修剪（REP），修剪集对树进行修剪，而不单单是选择。修剪集方法的优势在于生成树集，而非一个树。尤其是当领域专家算法对选择的树不满意时，可以从树集中进行人为挑选，树集归纳可以提高预测的精确度；将训练集分为两个子集的不足之处在于，设计决策受人为影响，小的训练集可能产生一个很小的树，因为低层部分被剪掉，但这恰恰是选择一个尽可能大的训练集的很好的理由，减小训练集的大小相当于增加了修剪预测精确度的不确定性。后期修剪有多种算法，如 MC-CP、REP、MEP、MDL、PEP、EBP 等，它们的精确度和简化程度各不相同。

下面介绍几种常用的后剪枝算法：

（1）减小错误剪枝（Reduced Error Pruning，REP）算法

REP 剪枝方法是由 Quinlan 提出的。在此方法中，检测决策树中非叶子的节点，若此节点被最佳的叶节点取代而产生的错误数目小于或者等于之前未修剪的决策树的错误数目，则修剪成功；否则修剪失败，放弃修剪。如果修剪成功，则重复这一方法直至错误数目比前一决策树有所增加为止。需要注意的是，此方法遵循的是由下而上的原则。

（2）悲观错误剪枝（Pessimistic Error Pruning，PEP）算法

PEP 方法也是由 Quinlan 在 1987 年提出的。Quinlan 发现，当用产生决策树的训练数据来检测错误率时，实际上对错误的估计过于乐观了。因为在大多数情况下，决策树是完全与训练数据相符合的，也就是说错误率为零。但当将一些与训练数据不同的数据应用于由训练数据而产生的决策树时，很明显这时的错误率将大大增加。基于以上原因，Quinlan 借用二项式分布中的连续修正对训练数据中的错误率加以修正，以得到更为符合实际的错误率。显然，与修正前的错误率相比，修正后的错误率增加了不少，因此认为它对错误率的看法是“悲观”的。悲观错误修剪法遵循由上而下的原则。

（3）最小编码长度（Minimum Description Length，MDL）算法

MDL 剪枝方法是 M. Mehta、J. Rissanen 和 R. Agrawal 等人在 1995 年提出的一种决策树后剪枝的方法，该方法将构造的决策树中所包含的信息量以二进制编码的形式来表示，利用编码的长度代表决策树某一分枝的误分类率的大小，进而决定剪枝与否，它的目的在于生成一棵描述长度最小的决策树。

（4）最小代价复杂度剪枝（Minimum Cost - Complexity Pruning，MCCP）算法

MCCP 剪枝算法在 1984 年被 Breiman 等人在著名的 CART 系统中使用。在该算法中，最下面的未被剪枝的节点成为树叶，并用它先前的分枝中最频繁的类进行标记。对于树中每一个非树叶节点，算法计算该节点上的子树被剪枝后可能出现的期望错误率。然后，使用每个分枝的错误率，结合每个分枝观察的权重评估，计算针对该节点剪枝的期望错误率。如果剪去该节点，导致较高的期望错误率，则保留该子树；否则剪去该子树。产生一组逐渐被剪枝的树后，使用一个独立的测试集评估每棵树的准确率，就能得到具有最小期望错误率的决策树。

以上介绍了几种主要的决策树剪枝的方法。当然，决策树剪枝的方法不止这些，各种不同的方法均有自己的特点。前剪枝方法简单，效率高。后剪枝比前剪枝需要更多的计算时间，但通常产生的决策树更为可靠。前剪枝方法和后剪枝方法也可以相互结合起来，从而构成一个混合的修剪方法。

2.7 决策树算法国内外研究现状

目前，在许多数据挖掘系统应用中，决策树技术已经得到了研究者和软件公司的极大关注，国外很多公司均推出了自己的数据挖掘系统，其中多数都采用了决策树方法。SAS 公司的 SAS Enterprise Miner 是一种通用的数据挖掘工具，能够通过收集分析各种统计资料和客户购买模式，帮助用户发现业务的趋势，解释已知事实，预测未来结果，并识别完成任务所需的关键因素，最终实现增加收入并降低成本的目的。IBM 公司的 Inteligent Miner 这个系统具有典型的数据集自动生成、关联发现、序列规律发现、概念性分类以及可视化显示等功能，在数据挖掘的过程中可以实现数据选取、数据转换、数据挖掘以及结果显示等一系列操作的自动化，是一款比较优秀的数据挖掘工具。RightPonit 公司开发的 Data-Cruncher 是一种基于客户/服务器模式的数据挖掘引擎。这个工具能够分析数据仓库中的海量数据，并且能够与许多主流的关系数据库和数据挖掘辅助工具进行直接的连接，进而能够辅助决策者建立面向营销的数据挖掘研究的模型。Isoft 公司的 AC2 是一个相当流行的决策树分析算法。Isoft 已与 Bussiness Objects 公司达成合作协议，根据协议，Bussiness Objects 公司将负责销售包含有 Isoft 决策树方法的数据挖掘模块。SPSS 公司向市场上销售的是一种基于 SI - CHAID 算法的数据挖掘产品。其他许多开发商则采用了将几种算法组合到一起的方法以增强其产品的性能。此外，数据挖掘系统目前市场上还有 SiliconGraphic 公司研究开发的 Minset，Vanguard Software 公司的 Decision Pro3.0，美国 Thinking Machine 公司推出的 Darwin 等。

决策树算法被广泛应用于多个领域，它经历了一个由浅到深、由简单到复杂的过程。分类研究在国外发展很快，已有很多成型的算法和模型，比较成熟的决策树算法有 ID3、C4.5、CART、PUBLIC、SLIQ、SPRINT 等。下面是国内外决策树算法发展过程中的一些

重要进展：

1979 年，J. R. Quinlan 提出的迭代分类器 ID3 算法是决策树分类算法的典型代表，也是决策树算法的理论基础算法。ID3 算法引入了信息论中熵的概念，在决策树生成过程中，采用信息增益作为决策树各级节点选择属性的标准，以便对每一个非叶子节点进行测试时，能获得关于被测试记录最大的类别信息。ID3 算法成为分类算法的基础，从此人们研究的大多数分类算法都是在 ID3 算法基础之上进行的改进算法。C4.5 算法就是其中最具典型的基于 ID3 算法基础之上的改进算法。1984 年，L. Breiman 等人提出了分类及回退树 CART 算法（即 Classification And Regression Tree 的缩写，分类回归树）。这个算法一经提出就得到了广泛的普及和推广，可以对连续型的变量进行处理。1993 年，Quinlan 出版的《机器学习规划》对 C4.5 算法进行了详细的描述。ID3 算法不能处理离散值属性数据和缺失数据，而 C4.5 算法弥补了 ID3 算法的不足之处。另外，C4.5 算法的重要改进是使用信息增益率来选择属性。1996 年，IBM 的 M. Mehta 和 R. Agrawal 等人提出了一种高速可伸缩地有监督地寻找学习 SLIQ 分类方法，它针对数据量远大于内存容量的情况采用了类表、属性表及类直方图三种数据结构。用换入换出策略处理大数据量。在建树阶段，对连续属性采取预排序技术与广度优先相结合的策略生成树，对离散属性采取快速的求子集算法确定划分条件。由于 SLIQ 的类表必须驻留在内存中，所以处理的数据量仍然有限。IBM 的 J. Shafer 在 1996 年提出了 SPRINT 算法，它通过属性表和类直方图两种数据结构对 SLIQ 能够处理的数据容量进行了进一步扩充，可以真正处理超大型数据库的数据。Rastogi 和 Shim 在 1998 年提出了 PUBLIC 算法，该算法提出了节点代价的目标函数，利用该函数计算节点代价的估计，如果能够确定某节点的分支将会在随后的修剪阶段剪掉，则停止其生长，从而建树和修剪阶段结合成为一个阶段，而不是依次执行这两个阶段，提高了决策树的执行效率。2003 年，Minos Garofalakis 等人提出了一种约束决策树（Decision Tree With Constraints），它对树的节点数和错误率分别设定了上界，采用了一种新颖的“分支 - 限制”方法对树的规模和精度加以控制，可以让生成的决策树具有节点更少，产生规则更易理解的优点。2007 年，Blanchard G 等人提出了一种改进的决策树算法，它的整体思想是基于优化二叉决策树（ODT）。这种方法对搜索过程进行了改进，并继承了传统方法的生成规则容易解释的特点，在效率方面有了一定的提高。

与国外相比，我国对数据挖掘的研究起步较晚，国内的不少学者也在决策树算法上进行了深入的研究，并且大多数都是在 ID3 算法的基础之上提出自己的改进思想，也取得了不少的成果。刘小虎博士和李生教授提出了 ID3 算法的优化算法——MID3 算法，以 ID3 为基础，提出改进的递归信息增益优化算法。每当选择一个新的属性时，算法不仅仅是考虑该属性带来的信息增益，还需要考虑到该属性后选择的属性带来的信息增益，即同时考虑树的两层节点。吴菲等人提出的二元决策树实现模型，采用遗传算法，选择构造二元决策树，并提出了相应的遗传算法——一种新的基于二叉决策树模型选择方法，例如使用趋势预测模型的验证，取得了较好的效果。郭茂祖博士和刘扬教授，针对这些缺陷的 ID3 太多取值的问题，提出了一种新的基于属性值的决策树的节点，进而导致 AVPI 归纳算法中的决策树的大小和测试速度都比 ID3 算法优异。杨宏伟博士和王熙照教授等均采用基于层次分解方法，通过产生多层次的决策树技术来解决问题。2005 年，黄定轩等提出一类加权连续属性的多变量决策树构造方法，首先利用粗糙集理论和模糊聚类理论确定连续多变量属性的选择问题，然后利用聚类中心算法建立等级标准中心以解决连续变量的区间划分问

题，其次将等价关系相对泛化的概念用于决策树中多变量检验的构造。2006 年，阳东升博士等通过对组织协作网与决策树的描述分析提出了组织结构设计的新思路——基于决策个体在任务上的协作关系设计最佳的决策树。另外，中科院计算技术研究所智能信息处理重点实验室开发的 MS Miner 是一种多策略知识发现平台，能够提供快捷有效的数据挖掘解决方案，提供多种知识发现方法。

2.8 ID3 算法

作为决策树算法中最早期的算法之一的 ID3 算法是由 Quinlan 于 1986 年提出的。它的前身是 1966 年由 Hunt 等人提出的概念学习系统 CLS（Concept Learning System）。CLS 算法的主要思想是从一个空的决策树出发，首先找出最有判别力的因素，然后把数据分成多个子集，每个子集又选择最有判别力的因素进行划分，一直进行到所有子集仅包含同一类型的数据为止。最后得到一棵决策树，可以用它来对新的样例进行分类。ID3 算法主要是引入了信息论中的互信息，即信息增益作为决策属性判别能力的度量，进行决策结点属性的选择。

在 ID3 算法中用信息增益作为属性选择的标准，使得对每个非叶子节点测试时，能获得关于被测试例子最大的类别信息。使用该属性将训练样本集分成子集后，系统的熵值最小，期望该非叶子节点到各个后代叶节点的平均路径最短，使生成的决策树的平均深度较小，从而提高分类的速度和准确率。ID3 算法的目标是使用一系列测试来将训练集迭代地划分为多个子集，使得每个子集中的对象尽量属于同一个类。

2.8.1 ID3 算法基本概念

ID3 算法的基本概念包括：

（1）决策树的每个内部节点对应样本的一个非类别属性，该节点的每棵子树代表这个属性的取值范围和一个子区间（子集）。一个叶节点代表从根节点到该叶节点的路径对应的样本所属的类别。

（2）决策树的每个内部节点都与具有最大信息量的非类别属性相关联，这规定什么是一棵好的决策树。

（3）通常用熵来衡量一个内部节点的信息量，熵的定义使用信息论中的定义。

定义 2.2　设 U 是论域，$\{X_1, \cdots, X_n\}$ 是 U 的一个划分，其上有概率分布 $p_1 = P(X_i)$，则称：

$$H(X) = -\sum_{i=1}^{n} P_i \log P_i \qquad \text{（公式 2-1）}$$

为信息源 X 的信息熵，其中对数取以 2 为底，而当某个 p_i 为零时，则可以理解为 $0 \cdot \log 0 = 0$。

定义 2.3　设 $Y = \begin{Bmatrix} Y_1 & Y_2 & \cdots & Y_n \\ q_1 & q_2 & \cdots & q_n \end{Bmatrix}$ 是一个信息源，即 $\{Y_1, \cdots, Y_n\}$ 是 U 的另一个划分，$p(Y_j) = q_j$，$\sum_{j=1}^{n} q_j = 1$，则已知信息源 X 是信息源 Y 的条件熵 $H(Y \mid X)$ 定义为：

$$H(Y \mid X) = \sum_{i=1}^{n} P(X_i) H(Y \mid X_i) \qquad \text{（公式 2-2）}$$

其中，$H(Y \mid X) = \sum_{i=1}^{n} P(Y_j \mid X_i) \log P(Y_j \mid X_i)$ 为事件 X_i 发生时信息源 Y 的条件熵。

在 ID3 算法分类问题中，每个实体用多个特征来描述，每个特征限于在一个离散集中取互斥的值。ID3 算法的基本原理如下：设 $E = F_1 \times F_2 \times \cdots \times F_n$ 是 n 维有穷向量空间，其中 F_i 是有穷离散符号集。E 中的元素 $e = \{V_1, V_2, \cdots, V_n\}$ 称为样本空间的例子，其中 $V_j \in F_i$，（$j = 1, 2, \cdots, n$）。为简单起见，假定样本例子在真实世界中仅有两个类别，在这种两个类别的归纳任务中，PE 和 NE 是实体分贝称为概念的正例和反例。假设向量空间 E 中的正、反例集的大小分别为 P、N，由决策树的基本思想，ID3 算法是基于如下两种假设实现的：

（1）在向量空间 E 上的一棵正确的决策树对任意样本集的分类概率同 E 中的正、反例的概率一致。

（2）根据定义（2-2），一棵决策树对一样本集做出正确分类，所需要的信息熵为：

$$I(P,N) = -\frac{P}{P+N}\log\frac{P}{P+N} - \frac{N}{P+N}\log\frac{N}{P+N} \qquad \text{（公式 2-3）}$$

如果选择属性 A 作为决策树的根，A 取 V 个不同的值 $\{A_1, \cdots, A_n\}$，利用属性 A 可以将 E 划分为 V 个子集 $\{E_1, \cdots, E_v\}$，其中 $E_i (1 \leqslant i \leqslant V)$ 包含了 E 中的属性 A 取 A_i 值的样本数据，假设 E_i 中含有 p_i 个正例和 n_i 个反例，那么子集 E_i 所需要的期望信息是 $I(p_i, n_i)$，以属性 A 为根，所需要的期望熵为：

$$E(A) = \sum_{i=1}^{V} \frac{p_i + n_i}{P+N} I(p_i, n_i) \qquad \text{公式（2-4）}$$

其中，$I(p_i, n_i) = \frac{p_i}{p_i + n_i}\log\frac{p_i}{p_i + n_i} - \frac{n_i}{p_i + n_i}\log\frac{n_i}{p_i + n_i}$。

以 A 为根的信息增益是：

$$Gain(A) = I(P,N) - E(A) \qquad \text{（公式 2-5）}$$

ID3 算法选择 $Gain(A)$ 最大的属性 A^* 作为根节点，对 A^* 的不同取值对应的 E 的 V 个子集 E_i 递归调用上述过程生成的 A^* 子节点 $B_1, \cdots, B_v$。

ID3 算法基本原理是基于两类问题，但是也很适用于多类问题。设样本集 S 共有 C 类样本，每类样本数为 p_i（$i = 1, 2, \cdots, C$）。如果以属性作为决策树的根，A 具有 V 个值 $A_1, \cdots, A_v$，它将 E 划分为 V 个子集 $E_1, \cdots, E_v$。假设 E_i 中含有的第 j 类样本个数为 P_{ij}（$j = 1, 2, \cdots, C$），那么子集 E_i 的信息量，即熵为：

$$H(E) = -\sum_{j=1}^{c} \frac{P_{ij}}{|E_i|}\log\frac{P_{ij}}{|E_i|} \qquad \text{公式（2-6）}$$

以 A 为根分类后的信息熵为：

$H(E \mid A) = \sum_{i=1}^{V} \frac{|E_i|}{|E|} \cdot H(E_i)$，选择属性 A^*，使 $H(E \mid A)$ 最小，信息增益最大。

在实际应用中，能正确分类训练样本集的决策树不止一棵。ID3 算法能得出节点最小的决策树。在 ID3 算法的每一个递归过程中，都对训练样本集进行查询以确定属性的信息增益，然而此时的工作只是查询样本的子集而没有对其分类。为了避免访问全部数据集，ID3 算法采用了成为窗口（Windows）的方法，窗口随机性是从数据集中选择一个子集。

采用该方法会大大加快构建决策树的速度。

ID3 算法的主算法如下：首先从训练样本集中随机选择一个窗口，也就是包含正例和反例的样本子集，用建树算法对当前窗口形成一颗决策树；其次，对训练集中的例子用所得到的决策树进行类别判定，找出错判的例子。若存在错判的例子，把他们插入窗口，转到建树过程，否则停止。其主要流程图如图 2－5 所示。

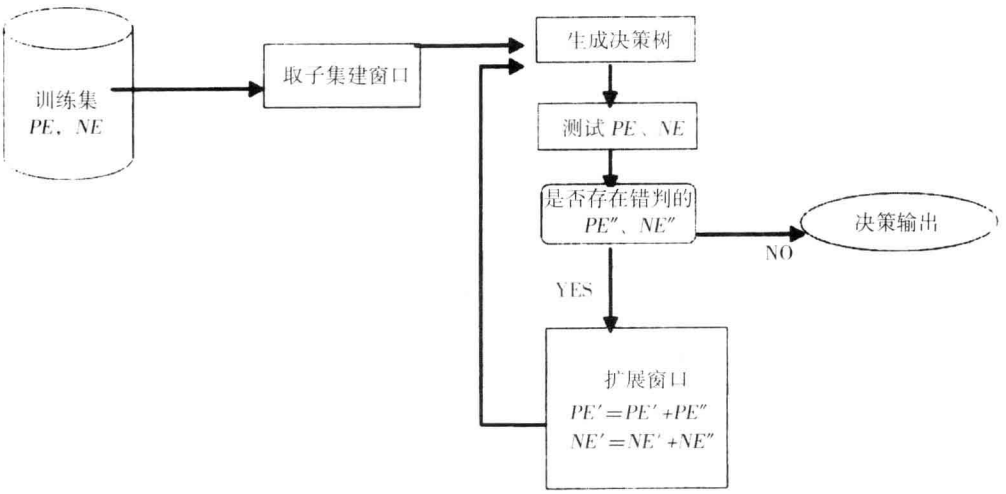


图 2－5　ID3 主算法流程图

其中 PE 、NE 分别代表正例集和反例集，它们共同组成训练样本集。PE'、PE'' 和 NE'、NE'' 分别代表正例集和反例集的子集。算法每迭代循环一次，生成的决策树将会不同。ID3 算法以一种从简单到复杂的爬山策略遍历这个假设空间，从空的树开始，然后逐步考虑更加复杂的假设。通过观察搜索空间和搜策策略，我们可以发现，它同样存在着一些优势和不足。

2.8.2　ID3 算法的优缺点

ID3 算法的优缺点可以总结如下：

（1）优点

ID3 算法在选择测试属性时使用了信息增益的概念，算法较容易实现而且算法的基础理论清晰。产生的决策树可以清晰地显示哪些字段比较重要，树的每个分枝都对应一个分类规则，因此产生的分类规则易于理解。ID3 算法构造的决策树平均深度较小，分类速度较快。

（2）缺点

ID3 算法不能够处理连续的属性值，如对连续值的属性，必须先对其离散化，不仅增加了分类算法的额外开销，还降低了分类的准确性。ID3 算法把信息增益作为分裂属性选择的标准，有偏向于属性取值数目较多的特征，而这一属性不一定是最优的。随着重复地将数据划分成越来越小的部分，ID3 决策树模型面临着较为严重的碎片问题。碎片是指一个给定分支中的样本数太小，没有统计意义。碎片问题带来的最直接的影响是决策树模型的健壮性降低，并最终降低分类准确率，一棵健壮的决策树所表现出来的特征之一是较少的叶子节点，即每个叶子节点有较多的样本数。

ID3 对训练样本集的质量的依赖性很强。训练样本的质量主要指是否存在噪声和是否存在足够的样本。

ID3 的决策树模型是多叉树，节点的子树个数取决于分支数据类型的不同取值个数，这不利于处理分支属性的取值数目较多的情况。目前流行的决策树算法大多采用二叉树模型。

ID3 算法不包含树的修剪，这样模型受噪声数据和统计波动的影响比较大。搜索中只能维持一个解，它可能往往是收敛于局部最优解而丢失全局的最优解，不能像候选剪除算法那样返回所有可能的与训练例集一致的假设，因此不容易使用新例步进式地改进决策树。

在不重建整棵树的条件下，不能方便地对判决树做更改。当一个新样本不能被正确分类时，就需要对树进行修改以适用于这一新样本。可以通过局部修补来实现这一点，但是这种情况下，决策树却会逐渐地失掉它作为某些重要概念的最有效表达的作用。

2.8.3 ID3 算法应用实例

根据上节对 ID3 算法的分析，下面用一个实例阐述 ID3 算法的建树过程。

表 2-1 是一个经典的例子。这个例子是根据报告天气条件的记录来决定是否适合打高尔夫球，类别属性是根据天气情况决定是否适合打高尔夫球。非类别属性有天气、气温、湿度、风。其中，天气属性值为：晴、多云、雨；气温属性值为：热、冷、适中；湿度属性值为：高、正常；风属性值为：无风、大风、中风。

表 2-1 训练样本集

天气	气温	湿度	风	适合打高尔夫球
多云	热	高	无风	不适合
晴	热	高	无风	适合
多云	热	高	大风	不适合
晴	热	高	中风	适合
雨	适中	高	无风	不适合
多云	热	高	中风	不适合
雨	热	正常	无风	适合
雨	适中	高	中风	不适合
雨	冷	正常	中风	不适合
晴	冷	正常	大风	适合
雨	热	正常	大风	不适合
晴	冷	正常	中风	适合
多云	适中	高	中风	不适合
多云	冷	正常	无风	适合
多云	适中	高	无风	不适合
雨	适中	正常	无风	不适合

（续表）

天气	气温	湿度	风	适合打高尔夫球
多云	冷	正常	中风	适合
雨	适中	正常	中风	不适合
多云	适中	正常	中风	适合
多云	适中	正常	大风	适合
晴	适中	高	大风	适合
晴	适中	高	中风	适合
晴	热	正常	无风	适合
雨	适中	高	大风	不适合

下面用 ID3 算法对表 2－1 中的训练集进行分类，具体计算过程如下：

（1）由公式（2－2），计算类别属性信息熵

根据表 2－1 所示，训练集共有 24 条记录，对于类别属性（适合打高尔夫球），其中类是正例（适合）的样本个数为 12 个，类是反例（不适合）的样本个数为 12 个，计算给定样本类别属性的信息熵为：

$$I(12,12) = -\frac{12}{24}\times\log\frac{12}{24}-\frac{12}{24}\log\frac{12}{24}=1$$

（2）由公式（2－3），计算非类别属性的信息熵

以非类别属性天气为例，其值取为晴、多云、雨。其中，取值晴的 7 个样本中正例的样本个数为 7 个，反例的样本个数为 0 个；取值多云的 9 个样本中正例的样本个数为 4 个，反例的样本个数为 5 个；取值雨的 8 个样本中正例的样本个数为 1 个，反例的样本个数为 7 个。可得天气属性的信息熵为：

$$E(\text{天气}) = \sum_{j=1}^{v}\frac{s_{1j}+s_{2j}}{s}I(s_{1j},s_{2j}) = \frac{7}{24}\left(-\frac{7}{7}\times\log\frac{7}{7}-0\right)+\frac{9}{24}\left(-\frac{4}{9}\times\log\frac{4}{9}-\frac{5}{9}\times\log\frac{5}{9}\right)+\frac{8}{24}\left(-\frac{1}{8}\times\log\frac{1}{8}-\frac{7}{8}\times\log\frac{7}{8}\right)=0.5528$$

同理，如果选择气温为测试属性，则气温属性信息熵为：

$$E(\text{气温}) = \sum_{j=1}^{v}\frac{s_{1j}+s_{2j}}{s}I(s_{1j},s_{2j}) = \frac{8}{24}\left(-\frac{4}{8}\times\log\frac{4}{8}-\frac{4}{8}\times\log\frac{4}{8}\right)+\frac{11}{24}\left(-\frac{4}{11}\times\log\frac{4}{11}-\frac{7}{11}\times\log\frac{7}{11}\right)+\frac{5}{24}\left(-\frac{4}{5}\times\log\frac{4}{5}-\frac{1}{5}\times\log\frac{1}{5}\right)=0.9172$$

如果选择湿度为测试属性，则湿度属性信息熵为：

$$E(\text{湿度}) = \sum_{j=1}^{v}\frac{s_{1j}+s_{2j}}{s}I(s_{1j},s_{2j}) = \frac{12}{24}\left(-\frac{4}{12}\times\log\frac{4}{12}-\frac{8}{12}\times\log\frac{8}{12}\right)+\frac{12}{24}\left(-\frac{8}{12}\times\log\frac{8}{12}-\frac{4}{12}\times\log\frac{4}{12}\right)=0.9183$$

如果选择风为测试属性，则风属性信息熵为：

$$E(\text{风}) = \sum_{j=1}^{v}\frac{s_{1j}+s_{2j}}{s}I(s_{1j},s_{2j}) = \frac{6}{24}\left(-\frac{3}{6}\times\log\frac{3}{6}-\frac{3}{6}\times\log\frac{3}{6}\right)+\frac{10}{24}\left(-\frac{5}{10}\times\log\frac{5}{10}-\frac{5}{10}\times\log\frac{5}{10}\right)+\frac{8}{24}\left(-\frac{4}{8}\times\log\frac{4}{8}-\frac{4}{8}\times\log\frac{4}{8}\right)=1$$

（3）计算属性的信息增益

$Gain$（天气）$= I(12, 12) - E$（天气）$= 1 - 0.5528 = 0.4472$

$Gain$（气温）$= I(12, 12) - E$（气温）$= 1 - 0.9172 = 0.0828$

$Gain$（湿度）$= I(12, 12) - E$（湿度）$= 1 - 0.9183 = 0.0817$

$Gain$（风）$= I(12, 12) - E$（风）$= 1 - 1 = 0$

由上述结果可得，$Gain$（天气）最大，即天气对分类的帮助最大，所以选择天气作为测试属性，即为决策树的根节点。

（4）用递归方法建立决策树

根据天气作为根节点将训练集分为三个子集，对三个子集依次递归使用上面的方法。当天气属性取值为晴时，7 个样本中都是正例，则根据 ID3 算法，得到一个叶子节点。当天气属性取值为多云时：

$$I(4, 5) = -\frac{4}{9} \times \log \frac{4}{9} - \frac{5}{9} \times \log \frac{5}{9} = 0.9911$$

如果选择气温为测试属性，则：

$$E(\text{气温}) = \frac{3}{9}\left(0 - \frac{3}{3} \times \log \frac{3}{3}\right) + \frac{4}{9}\left(-\frac{2}{4} \times \log \frac{2}{4} - \frac{2}{4} \times \log \frac{2}{4}\right) + \frac{2}{9}\left(-\frac{2}{2} \times \log \frac{2}{2} - 0\right) = 0.4444$$

同理，如果选择湿度为测试属性，则

$$E(\text{湿度}) = \frac{5}{9}\left(0 - \frac{5}{5} \times \log \frac{5}{5}\right) + \frac{4}{9}\left(-\frac{4}{4} \times \log \frac{4}{4} - 0\right) = 0$$

同理，如果选择风为测试属性，则

$$E(\text{风}) = \frac{2}{9}\left(-\frac{1}{2} \times \log \frac{1}{2} - \frac{1}{2} \times \log \frac{1}{2}\right) + \frac{4}{9}\left(-\frac{2}{4} \times \log \frac{2}{4} - \frac{2}{4} \times \log \frac{2}{4}\right) + \frac{3}{9}\left(-\frac{1}{3} \times \log \frac{1}{3} - \frac{2}{3} \times \log \frac{2}{3}\right) = 0.9728$$

由公式（2-5）：

$$Gain(\text{气温}) = 0.9911 - 0.4444 = 0.5466$$

$$Gain(\text{湿度}) = 0.9911 - 0 = 0.9911$$

$$Gain(\text{风}) = 0.9911 - 0.9728 = 0.0183$$

由此可见，$Gain$（湿度）最大。则当天气取多云时，应选择湿度作为测试属性，成为下一个节点。该节点又分为二个子集，当湿度属性值为高时，共 5 个样本均为反例（不适合），则生成一个叶节点；当湿度属性值为正常时，共 4 个样本均为正例（适合），也生成一个叶节点。

当天气属性值为雨时，8 个样本中有 7 个反例，1 个正例。由公式（2-3）：

$$I(1, 7) = -\frac{1}{8} \times \log \frac{1}{8} - \frac{7}{8} \times \log \frac{7}{8} = 0.5436$$

如果选择气温为测试属性，则由公式（2-4）：

$$E(\text{气温}) = \frac{2}{8}\left(-\frac{1}{2} \times \log \frac{1}{2} - \frac{1}{2} \times \log \frac{1}{2}\right) + \frac{5}{8} \times 0 + \frac{1}{8} \times 0 = 0.2500$$

同理：如果选择湿度为测试属性，则：

$$E(\text{湿度}) = \frac{3}{8} \times 0 + \frac{5}{8}\left(-\frac{1}{5} \times \log\frac{1}{5} - -\frac{4}{5} \times \log\frac{4}{5}\right) = 0.4512$$

同理：如果选择风为测试属性，则：

$$E(\text{风}) = \frac{2}{8} \times 0 + \frac{3}{8} \times 0 + \frac{3}{8}\left(-\frac{1}{3} \times \log\frac{1}{3} - \frac{2}{3} \times \log\frac{2}{3}\right) = 0.3444$$

由上述计算可知：E（气温）最小，则 $Gain$（气温）最大。则当天气属性值取雨时，应选择气温为测试属性。

递归使用上述方法，可得到一棵由 ID3 算法构造的决策树，如图 2-6 所示。

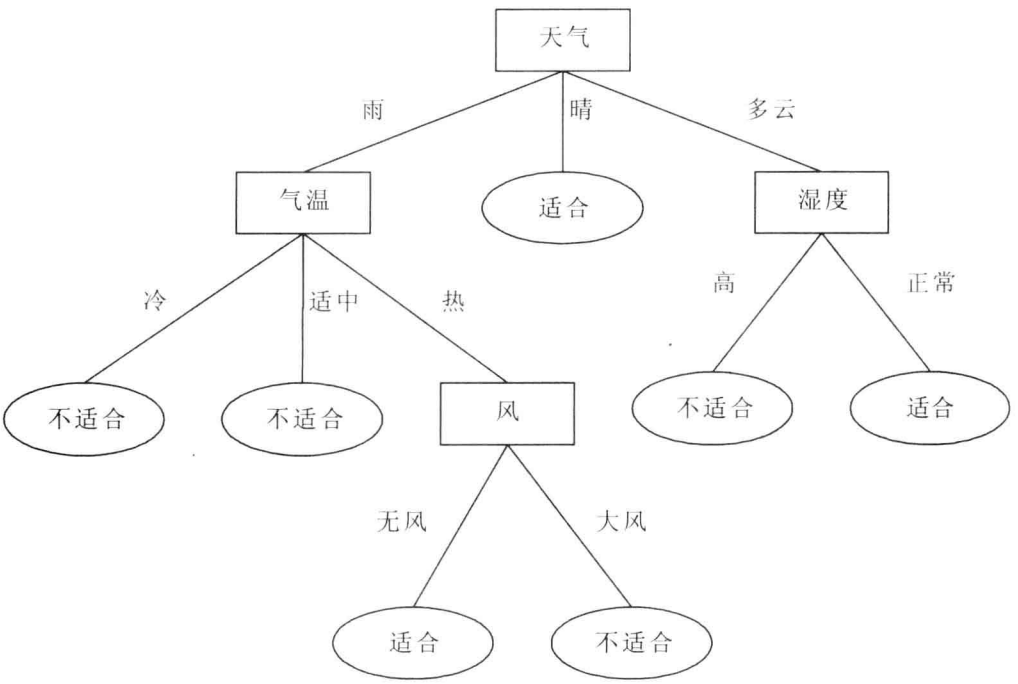


图 2-6　ID3 算法生成的决策树

2.9　C4.5 算法

上节已经提到，ID3 算法存在着许多需要改进的地方，为此，J. R. Quinlan 于 1993 年在其《Programs for Machine Learning》一书中，对 ID3 算法进行了补充和改进，提出了 C4.5 算法。在应用单机的决策树算法中，C4.5 算法不仅分类准确率高而且速度快。它与 ID3 算法的不同点总结如下：

（1）分支指标采用增益比率，而不是 ID3 所使用的信息增益。

（2）按照数值属性值的大小对样本排序，从中选择一个分割点，划分数值属性的区间，从而将 ID3 处理能力扩充到数值属性上来。

（3）将训练样本集中的未知属性用常用的值代替，或者用该属性的所有取值的平均值

代替，从而处理缺少属性值的训练样本。

（4）使用 K 次迭代交叉验证，评估模型的优劣程度。

（5）根据生成的决策树，可以产生一个 IF－THEN 规则集合，每一个规则代表从根节点到叶节点的一条路径。

2.9.1 熵和信息增益率

设数据划分 D 为类标记的元组的训练集。假定类标号属性有 m 个不同值，定义 m 个不同的 C_i 类（$i=1, \cdots, m$）。设 $C_{i,d}$ 是 D 中 C_i 类的元组的集合，$|D|$ 和 $|C_{i,d}|$ 分别是 D 和 $C_{i,d}$ 中元组的个数。

对 D 中的元组分类所需的期望信息为：

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i) \qquad \text{（公式 2－7）}$$

p_i 是 D 中任意元组属于类 C_i 的概率，并用 $|C_{i,d}|/|D|$ 估计。$Info(D)$ 又称为 D 的熵（Entropy）。

假设用属性 A 划分 D 中的元组，其中属性 A 根据训练数据的观测具有 v 个不同值 $\{a_1, a_2, \cdots, a_v\}$。可以用属性 A 将 D 划分为 v 个子集 $\{D_1, D_2, \cdots, D_v\}$，其中 D_j 包含 D 中的元组，它们在 A 上具有值 a_j。那么属性 A 的信息熵为：

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j) \qquad \text{（公式 2－8）}$$

信息增益定义为原来的信息需求与新的需求（即对 A 划分后得到的）之间的差。即：

$$Gain(A) = Info(D) - Info_A(D) \qquad \text{（公式 2－9）}$$

属性 A 有 v 个不同的值 $\{a_1, a_2, \cdots, A_v\}$，可以用属性 A 将 D 划分为 v 个子集 $\{D_1, D_2, \cdots, D_v\}$，其中 D_i 包含 D 中这样一些样本：它们在 A 上具有 A_i。假如我们以属性 A 的值为基准对样本进行分割，$SplitInfo_A$ 就是初始信息量：

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right) \qquad \text{（公式 2－10）}$$

信息增益率定义为信息增益与初始信息量的比值：

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)} \qquad \text{（公式 2－11）}$$

采用信息增益率作为选择属性的标准，克服了信息增益度量的缺点，但是算法偏向于选择值较集中的属性，而它并不一定是对分类最重要的属性。

算法流程图如图 2－7 所示。

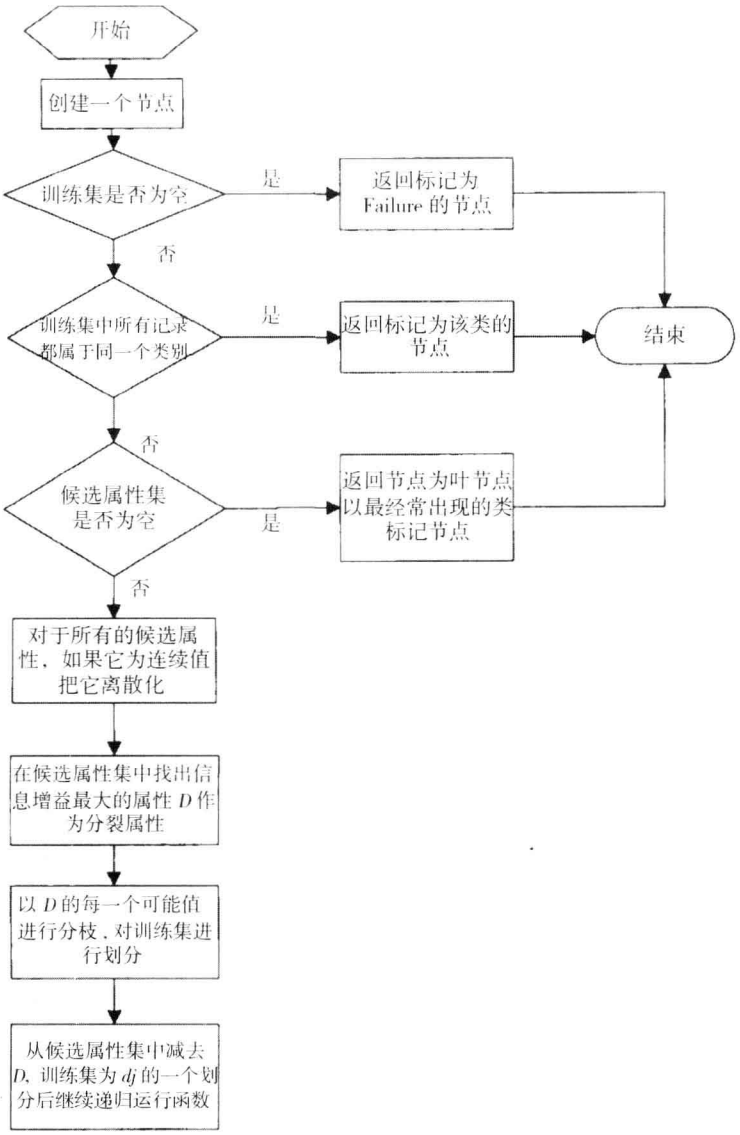


图 2－7　C4.5 算法流程图

2.9.2 数值属性的处理

（1）对连续属性的处理

在 ID3 算法的基础上，C4.5 算法在 ID3 算法的基础上增加了对连续型属性和属性值空缺情况的处理，对树剪枝和生成规则也有了较成熟的方法。C4.5 算法不仅可以处理离散属性，还可以处理连续属性。ID3 算法最初假定属性为离散值，但在实际环境中，很多属性值是连续的，C4.5 处理过程如下：

①根据属性的值，对数据集排序；

②用不同的阈值将数据集动态地进行划分；

③当输出改变时确定一个阈值；

④取两个实际值的中点作为一个阈值；

⑤取两个划分，所有样本都在这两个划分中；

⑥得到所有可能的阈值、增益及增益率；

⑦每一个属性会变为两个取值，即小于阈值或大于等于阈值。

（2）对未知属性值（取值为空）的处理

考虑到一些样本的某些属性取值可能为空，C4.5 采用如下的方法处理在计算期望信息熵时，只计算那些已知测试属性值的样本，然后乘以这些样本在当前结点的训练集中的比例作为整个训练集的期望信息熵；在计算划分信息熵时，把那些丢失测试属性值的样本作为一个新的类别对待，单独计算这些样本的期望信息熵；在划分训练集时，先将正常的样本按照一般的算法划分成几个子集，然后把那些丢失测试属性值的样本按照一定的概率分布到各个子集中。子集中的丢失测试属性值的样本与有测试属性值的样本保持一个比例关系：在对丢失测试属性值的未知事例分类时，C4.5 将该事例通过所有的分支，然后将结果进行合并，使它成为在类上的概率分布而不是某一个类。最终结果是具有最大概率的类。

2.9.3 C4.5 算法的缺点

（1）C4.5 采用的是分而治之的策略，在构造树的内部结点的时候是局部最优的搜索方式，所以它所得到的最终结果尽管有很高的准确性，仍然达不到全局最优的结果。

（2）C4.5 评价决策树最主要的依据是决策树的错误率，对树的深度、结点的个数等不进行考虑，而树平均深度直接对应着决策树的预测速度，树的结点个数则代表树的规模。

（3）一边构造决策树，一边进行评价，决策树构造出来之后，很难再调整树的结构和内容，决策树性能的改善十分困难。

（4）C4.5 在进行属性值分组时逐个试探，没有一种使用启发搜索的机制，分组时的效率较低。

（5）信息增益率函数也有它的缺陷。如果划分的优度非常小，也就是划分的信息熵值非常小，信息增益率会不稳定。当属性 a 只有一个属性值，或者训练实例对属性 a 只有一个取值的时候，$SplitInformation(S, a)$ 的取值为零。此时或者使得信息增益率没有意义，或者使得决策树进行一次没有意义的分类（即决策树的这个内结点只有一个子结点）。

2.9.4 C4.5 算法应用实例

（1）数据准备和数据预处理

银行在以往的信贷业务中，积累了大量客户信息和还贷情况等数据，确定是否对具备贷款担保条件的客户发放贷款，存在决策分析问题。

在选择哪些客户属性作为决策属性时，可以去掉一些对贷款风险无关的属性，如姓名、性别、身份证号、联系方式等信息，选出的属性有客户的年收入、有无住房、信誉度、贷款期限和是否拖欠贷款 5 个属性，其中是否拖欠贷款是我们要进行测试的目标属性，建立判断客户是否拖欠贷款的训练样本集，如表 2-2 所示。

表 2－2　银行信贷训练样本集

编号	年收入	有无住房	信誉度	贷款期限	是否拖欠贷款
1	≥5 万	有	一般	中期	否
2	<3 万	有	一般	中期	是
3	≥5 万	有	良好	短期	否
4	3 万～5 万	无	一般	中期	是
5	3 万～5 万	有	良好	短期	否
6	<3 万	无	一般	短期	是
7	3 万～5 万	有	良好	中期	否
8	≥5 万	有	一般	中期	否
9	<3 万	有	良好	短期	否
10	3 万～5 万	无	良好	短期	否

（2）构造决策树

根据公式 2－7～公式 2－10 对客户信贷情况的数据进行计算，得到 4 个属性的信息增益和信息增益率，如表 2－3 所示。

表 2－3　客户信贷业务中的属性信息增益和信息增益率

变量值	年收入	有无住房	信誉度	贷款期限
信息增益	0.2814	0.1917	0.3959	0.0319
信息增益率	0.1791	0.2175	0.3959	0.0319

从表 2－3 可知，属性信誉度具有最大的信息增益率值，故选择该属性作为决策树的第一个分节点进行测试。在构造决策树的过程中以是否拖欠贷款作为目标属性，根据决策树 C4.5 算法和表 2－3 生成的决策树，并对决策树进行剪枝，修剪后的决策树如图 2－8 所示。

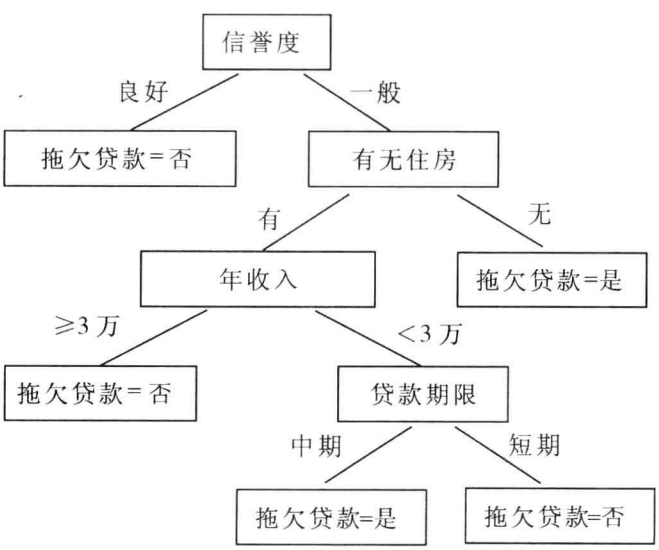


图 2－8　修剪后的决策树

(3) 提取决策树分类规则

基于图 2-8 的决策树，对其从根到树叶的每条路径创建一个规则，以 IF-THEN 形式的分类规则表示。该分类沿着给定路径上的每个属性和属性相关联的值形成规则前件（“if”）的一个合取项，则叶节点包含类预测，形成后件（“then”）部分，得出最后结论。从图 2-8 中提出的规则如下：

规则 1：如果信誉度良好，则该客户不拖欠贷款。

规则 2：如果信誉度一般，并且无住房，则该客户拖欠贷款。

规则 3：如果信誉度一般，并且有住房、年收入≥3 万，则该客户不拖欠贷款。

规则 4：如果信誉度一般，并且有住房、年收入 <3 万并且贷款期限是中期，则该客户拖欠贷款。

规则 5：如果信誉度一般，并且有住房、年收入 <3 万并且贷款期限是短期，则该客户不拖欠贷款。

我们从这个决策树模型可以看出，影响客户能够按时偿还贷款的属性主要由信誉度决定，如果信誉度良好，客户都会按时偿还贷款。有了这个预测模型，银行就可以根据该预测模型，判断申请银行贷款的客户能否按时还贷款。例如一个客户，信誉度等级为一般，年收入为 2 万，贷款期限是短期，则根据预测模型可知该客户是不会拖欠贷款的，应该准予贷款。这样就能大大降低银行信贷业务中的风险，使银行的信贷风险尽可能降到最低，从而减少银行的经济损失。

2.10 CART 算法

CART 算法是 Classification And Regression Tree 的简称，可以处理高度倾斜或多态的数值型数据，也可处理顺序或无序的类属型数据。这种算法与 ID3、C4.5 算法最大的不同就是选择具有最小基尼指数值的属性作为测试属性。该算法采用一种二分递归分割的技术，将当前样本集分为两个子样本集，使得生成的决策树的每个非叶子节点都有两个分枝。因此，CART 算法生成的决策树是二叉树。

另外，与 ID3、C4.5 的算法只为叶子节点分配类别不同，CART 算法考虑到每个节点都有成为叶子节点的可能，对每个节点（包括叶子节点和非叶子节点）都分配类别。分配类别的方法可以用当前节点中出现最多的类别，也可以参考当前节点的分类错误或其他方法。

CART 算法采用一种称为交叉确定的剪枝方法。该方法解决了在小样本集上挖掘决策树由于没有独立测试样本集而造成的过度拟合问题。

定义 2-3：设样本集中包含各类，则 $gini(T) = 1 - \sum_{j=1}^{n} P_j^2$，其中 P_j 是 T 中包含类 j 的概率。若将 T 划分为两个子集 T_1 和 T_2，则，$gini(T_1, T_2) = \frac{T_1}{T} gini(T_1) + \frac{T_2}{T} gini(T_2)$。

与 C4.5 算法类似，CART 算法也是先建树后剪枝，但是在具体实现上有所不同。由于二叉树不容易产生数据碎片，精确度往往高于多叉树，因此 CART 算法采用 2 分递归划分，在分支节点上进行布尔测试，判断条件为真的划归为左分支，否则划归为右分支，最终形成一棵二叉决策树。对于连续属性 A，判断 $A \leqslant V$ 是否成立；对于离散型的属性 A，

判断 $A \in S'$ 是否成立，包括属性 A 所有取值的子集，可以用贪心算法或穷举法确定。CART 算法在进行建树过程中，不管节点 N 是否被划分，均给标记相应的类，方法是判断不等式。

$$\frac{C(j \mid i)\,\Pi(i)N_i(t)}{C(i \mid j)\,\Pi(j)N_j(t)} > \frac{N_i}{N_j} \qquad \text{公式（2-12）}$$

若对于除 i 以外的所有类 j 都成立，则将 N 标记为类 i。其中，$\Pi(i)$ 表示类 i 的先验概率，N_i 是训练样本集中类 i 的数量，$N_i(t)$ 是节点 N 的样本中类 i 的数量。$C(j \mid i)$ 表示将 i 错误分类为 j 的代价，可通过查找决策损失矩阵得到。CART 算法在满足下述条件之一时停止建树：（1）所以叶节点中的样本数为 1 或者样本属于同一类。（2）决策树高度到达用户设计的阀值。

CART 算法仍然使用后剪枝。在树生成的过程中，多展开一层就会有多一些的信息被发现，CART 算法运行到不能再长出分枝为止，从而得到一棵最大的决策树。然后 CART 对这棵超大的决策树进行剪枝。剪枝算法通过使用独立于训练样本集的测试样本集对子树的分类错误进行计算，找出分类错误最小的子树作为最终的分类模型。有些样本集，由于样本数太少而不能分出独立的测试样本集，CART 算法采用一种称为交叉确定（Crosvalidation）的剪枝方法。该方法将原始训练样本集分成 N 份（假定每份的数据分布近似或相同）。N 份中取第 1 份作为测试集，其余 $N-1$ 份合并后用于训练集，经过一次完整的建树和剪枝的过程，得到一棵局部决策树。以此类推，直到 N 份样本集都做了一次测试集为止。这样总共得到 N 棵局部决策树，综合这 N 棵局部决策树使之形成全局决策树。这样形成的全局决策树在性能上非常近似于由包含所有样本的原始训练样本集得出的决策树。该方法解决了在小样本集上挖掘决策树时由于没有独立测试样本集而造成的过度拟合问题。

2.11 SLIQ 算法

上述算法由于要求训练样本驻留内存，因此不适合处理大规模数据。为此，IBM 研究人员提出了一种快速的、可伸缩的、适合处理较大规模数据的决策树分类算法 SLIQ（Supervised Learning In Quest）。该算法利用 3 种数据结构来构造树，分别是属性表、类表和类直方图。属性表含有两个字段：属性值和样本号。类表也含有两个字段：样本类别和样本所属叶节点。类表的第 k 条记录对应于训练集中的第 k 个样本（样本号为 k），所以属性表和类表之间可以建立关联。类表可以随时指示样本所属的划分，所以必须长驻内存。每个属性都有一张属性表，可以驻留磁盘。类直方图附属在叶节点上，用来描述节点上某个属性的类别分布。描述连续属性分布时，它由一组二元组 < 类别，该类别的样本数 > 组成；描述离散属性分布时，它由一组三元组 < 属性值，类别，该类别中取该属性值的样本数 > 组成。随着算法的执行，类直方图中的值不断更新。SLIQ 算法在建树阶段，对连续属性采取预排序技术与广度优先相结合的策略生成树，对离散属性采取快速的求子集算法确定划分条件。

具体步骤如下：

（1）建立类表和各个属性表，并且进行预排序，即对每个连续属性的属性表进行独立排序，以避免在每个节点都要给连续属性值重新排序。

（2）如果每个叶节点中的样本都能归成一类，则算法停止；否则转（3）。

（3）利用属性表寻找拥有最小 *gini* 值的划分作为最佳划分方案。算法一次只处理一张属性表（假设对应于属性 A），从上往下每读一条记录，就根据样本号关联到类表的相关记录，找到样本所在的叶节点，从而更新叶节点上的类直方图；若 A 是连续的，则还要根据 A 的当前取值 v 计算对应于判断 $A \leqslant v$ 的 *gini* 值。若 A 是离散的，则在扫描完 A 属性表后，用贪心算法或穷举法计算最佳的。当扫描结束时，就可确定在各叶节点上依据属性 A 进行划分的最佳方案，各叶节点上的划分方案不一定相同。同理，当扫描完所有属性表时，可确定当前树中所有叶节点的最佳划分。

（4）根据（3）步得到的最佳方案划分节点，判断为真的样本划归左孩子节点，否则划归右孩子节点。这样，（3）、（4）步就构成了广度优先的生成树策略。

（5）更新类表中的第二项，使之指向样本划分后所在的叶节点。

（6）转（2）。

SLIQ 算法采取基于 MDL（最小描述长度）原则进行剪枝，即根据决策树的编码代价的大小进行剪枝。剪枝的目标就是寻找最小代价树。

SLIQ 定义：决策树编码代价 = 树结构的编码代价 + 在内部节点上进行测试的编码代价 + 训练集的编码代价。树结构的编码代价等于树中各节点 t 的编码代价 $L(t)$ 之和，而 $L(t)$ 按如下情况进行定义：

（1）若树中只含有叶节点或度为 2 的节点，只需 1bit 区分节点，则 $L(t)$ 为 1；

（2）若树中含有叶节点、只有左孩子的节点、只有右孩子的节点及度为 2 的节点，需 2bit 区分节点，则 $L(t)$ 为 2；

（3）若树中仅考虑分支节点，即只有左孩子的节点、只有右孩子的节点及度为 2 的节点，需 log3 个 bit 区分节点，则 $L(t)$ 为 log3。在内部节点上进行测试的编码代价 *Ltest* 与测试属性的类型有关。若测试属性是连续属性，则 *Ltest* 定义为 1；若测试属性是离散属性 A，设在决策树中用 A 进行测试的次数为 S，则 *Ltest* 定义为 $\ln_s$。训练集的编码代价定义为给训练集分类得到的所有分类错误之和 *Errors*。

根据上述定义，每个节点 t 的编码代价 $C(t)$ 的计算方法如下：

（1）如果 t 为叶节点，则 $C(t) = L(t) + Errors(t)$；

（2）如果 t 含有两个孩子 t_1、t_2，则 $C(t) = L(t) + Ltest + C(t_1) + C(t_2)$；

（3）如果 t 只有左孩子 t_1，则 $C(t) = L(t) + Ltest + C(t_1) + C'(t_2)$；

（4）如果 t 只有右孩子 t_2，则 $C(t) = L(t) + Ltest + C(t_2) + C'(t_1)$。

剪枝一般有 3 种策略，分别是完全剪枝、部分剪枝和混合剪枝。对于某个分支节点，完全剪枝只考虑上述（1）、（2）清况，即要么不剪枝，要么将该节点的两个孩子全部剪去；部分剪枝考虑上述 4 种情况，从而决定不剪枝、剪去节点的两个孩子，还是只剪去其中一个孩子，其中（3）、（4）中的 $C'(t_i)$ 表示被剪枝孩子中的样本留在父节点 t 中而导致的分类错误；混合策略分别使用前两种策略进行分步剪枝。选定一种剪枝策略后，从叶节点向上，为每个节点 t 计算不同情况时的 $C(t)$，再根据最小的 $C(t)$ 所对应的情况进行剪枝。实践证明，对于前面算法可以处理的小规模训练集，SLIQ 的运行速度更快，生成的决策树更小，预测的精确度较高；对于前面算法无法处理的大型训练集，SLIQ 的精确度更高，优势更明显。

2.12 SPRINT 算法

SPRINT 是对 SLIQ 的改进算法，其目的有两个：一是为了能够更好地并行地建立决策树；二是为了适合更大的数据集。SPRINT 修改了 SLIQ 的数据结构，SPRINT 不具有类列表，只具有属性列表，属性列表的每一行包括如下三个部分：属性值、对应的类别和样本的记录号。当一个节点分裂成 2 个子节点时，属性列表也分裂成两个列表。因为连续属性列表是预先排序的，所以分裂后的列表依然是有序的。SPRINT 不具有类列表，所有的属性列表都可以换入或换出内存，对数据集的大小，SPRINT 几乎可以说是没有限制。

SPRINT 对数据结构如此修改，除了能够使决策树生成算法不受数据集大小的限制，并且速度更快，还能使可伸缩性更好并易于实现并行计算。

SLIQ 算法不要求所有的训练数据都驻留在内存，这对于那种因为属性数目非常多而不能常驻内存的训练集非常有效。但是，SLIQ 算法在运行过程中要频繁地访问和更新类表，为了提高效率，必须使类表常驻内存中。类表的规模随着训练样本的个数的增加而线性增长，这也就限制了 SLIQ 的扩展能力。当训练集增加导致类表放不进内存时，算法就无法进行，这限制了 SLIQ 处理数据的最大规模。为此，IBM 研究人员提出可伸缩、可并行化的决策树算法 SPRINT，它消除了所有内存限制，运行速度快，且允许多个处理器协同创建一个决策树模型。

SPRINT 定义了两种数据结构，分别是属性表和直方图。直方图附属在节点上，用来描述节点上某个属性的类别分布。当描述连续属性的类分布时，节点上关联两个直方图 *Cbelow* 和 *Cabove*，前者描述已处理样本的类别分布，后者描述未处理样本的类别分布，两者的值皆随算法进行而更新；当描述离散属性的类分布时，节点上只关联一个直方图 *Count Matrix*。与 SLIQ 算法不同，SPRINT 算法采取传统的深度优先生成树策略，具体步骤如下：

（1）生成根节点，并为所有属性建立属性表，同时预排序连续属性的属性表。

（2）如果节点中的样本可归成一类，则算法停止；否则转（3）。

（3）利用属性表寻找拥有最小 *gini* 值的划分作为最佳划分方案。算法依次扫描该节点上的每张属性表。由于计算 *gini* 值需要该节点上直方图的信息，因此处理每一张属性表前均要清空并初始化直方图。对于连续属性 A，要求初始化 *Cbelow* 为 0、*Cbove* 为该节点上的样本关于 A 的总体分布。每扫描 A 属性表的一条记录（设属性值为 v），都要更新一次 *Cbelow* 和 *Cbove*，且计算对应于划分 $A \leqslant v$ 的 *gini* 值。这样，一遍扫描完 A 属性表，就可得到该节点上关于 A 属性的最佳划分，并记录下来。对于离散属性 B，先清空 *Count Matrix*，在扫描 B 属性表的同时更新 *Count Matrix*，扫描完后，根据构造好的 *Count Matrix* 求解。最佳划分中的 S，并记录下来。当扫描完该节点的所有属性表时，就可得到该节点的最佳划分方案。

（4）根据划分方案，生成该节点的两个子节点 $N1$、$N2$。

（5）划分该节点上的各属性表，使之关联到 $N1$ 或 $N2$ 上。首先划分测试属性的属性表：依次扫描记录 < 属性值 x，类别属性 c，样本号 id >，判断 x 属于哪个孩子，然后将该条记录移至该孩子节点的新属性表中，同时将记值写入 Hash 表，以保存划分后样本所在

孩子节点的信息。然后划分其余属性表：依次扫描记录，依据样本号查找 Hash 表中该样本划分后所在的孩子节点，然后将该记录移至该孩子节点的新属性表中。

（6）分别转（2）考察 $N1$、$N2$ 节点。

SPRINT 算法在剪枝阶段进行基于 MDL 的剪枝，至此构成了 SPRINT 的串行化算法。将串行化算法稍加改进，就成为并行化算法：将训练集平均分布到 N 个处理器上，而后每个处理器生成自己的数据分片。由于连续属性值要求全局排序，因此要将 N 个处理器上的连续属性的属性表综合重排序，再平均分布到 N 个处理器上。在建立 Hash 表之前，需要从 N 个处理器上收集所有的样本号信息。

实践表明，在 SLIQ 的类表可以存进内存的情况下，SPRNIT 比 SLIQ 执行得慢；然而在训练集规模超过 SLIQ 能承受的最大规模后，SPRINT 的效率比 SLIQ 的要好得多，且数据量越大，SPRINT 的效率越高。

SPRINT 分类方法最大的优点就是可以避免内存空间的限制，利用多个并行处理器构造一个稳定的、分类准确率很高的决策树，具有很好的可伸缩性、扩容性，但该算法因使用属性列表，使得存储代价大大增加，并且节点分割处理的过程较为复杂，加大了系统的负担。

2.13 PUBLIC 算法

PUBLIC（Pruning and Building Integrated in Classification）算法将建树、剪枝结合到一步完成，在建树阶段不生成会被剪枝的子树，因而大大提高了效率。PUBLIC 的建树基于 SPRINT 方法、剪枝基于 MDL 原则，但 MDL 原则不能直接应用。因为生长不完全的树提供的信息不完全，所以对于仍要扩展的叶节点，估算的编码代价偏高，这会导致过度剪枝，从而降低预测精确度。

PUBLIC 采用低估策略来矫正过高的代价估算，即对将要扩展的叶节点计算编码代价的较低阈值，而对于另两类叶节点（剪枝形成的、不能被扩展的），估算方法不变。计算较低阈值的方法有多种，但最简单有效的方法是将它设置为 1。另外，PUBLIC 对决策树编码代价的计算方法不同于 SLIQ。PUBLIC 在建树过程中定期调用剪枝算法，避免生成被剪枝的节点，因而节省了开销，提高了效率。

上述含有剪枝的算法都是分成两步进行，即先建树再剪枝。PUBLIC 算法将建树、剪枝结合到一步完成，在建树阶段不生成会被剪枝的子树，因而大大提高了效率。PUBLIC 对节点 N 进行判断剪枝的步骤如下：

（1）if（N 是将被扩展的叶节点）then 返回 N 的较低闭值 cost1；

（2）if（N 是剪枝形成的或不可扩展的叶节点）then 返回 N 的编码代价 cost2；

（3）eles {计算 N 的编码代价 cost3；//N 是含有两个子节点 N1 和 N2 的分支节点；

（4）if（cost2 < cost3）then {将 N1 和 N2 剪枝；N 变为剪枝后的叶节点}；

（5）返回 cost2 与 cost3 中的较小值。

2.14 决策树算法比较

以上介绍的是决策树算法中典型的六种算法，其实基于决策树的分类算法种类繁多，各种算法在执行速度、输出结果的可理解性以及可扩展性，分类预测的准确性等方面都各有千秋。ID3 算法是最具有代表性、最核心的算法，采用信息论中的信息原理来选择属性分割数据集，但是该算法存在一个严重的缺点，就是只能处理离散属性的并且属性值不遗漏的数据样本。继而出现了 C4.5 算法，对原来的 ID3 算法进行了改进，能够直接处理连续属性的数据，也能够处理属性值空缺、有噪声的数据样本集。ID3 算法和 C4.5 算法生成的决策树分支会随之属性值的增加而变得很庞大，即生成多叉树。而 CART 算法使用了基尼系数来选择属性，生成的是结构清晰、容易转换成规则的二叉树。随着网络的盛行，决策树算法的可扩展性和并行性被提出，比较有代表的是 SLIQ 和 SPRINT 算法。这些算法在每个方面都有自己的特点和优劣势，使用的关键技术也各不相同，在具体使用时要根据具体情况使用。

表 2-4 为决策树中的六类算法在决策树结构、选择测试属性的技术、对连续属性的处理、剪枝的执行效率、是否必须独立测试样本集以及可伸缩性、并行性等方面进行的比较。

表 2-4 几种决策树方法比较

	选择测试属性的技术	连续属性的处理技术	剪枝方法	是否必须独立测试样本	可伸缩性	并行性	决策树的结构
ID3	信息增益	离散化	分类错误	是	差	差	多叉树
C4.5	信息增益率	预排序	分类错误	否	差	差	多叉树
CART	*gini* 系数	预排序	分类错误	否	差	差	二叉树
SLIQ	*gini* 系数	预排序	MDL	否	良好	良好	二叉树
SPRINT	*gini* 系数	预排序	MDL	否	好	好	二叉树
PUBLIC	*gini* 系数	离散化	MDL	否	差	差	二叉树

第 3 章　贝叶斯分类算法

贝叶斯，英国数学家，神父，他于 1702 年出生于伦敦，1742 年成为英国皇家学会会员，1763 年 4 月 7 日逝世。贝叶斯在数学方面主要集中于研究概率论方面的问题。他首先将归纳推理法用于概率论基础理论，并且创立了贝叶斯统计理论，在统计决策函数、统计推断、统计的估算等各个研究领域作出了卓越的贡献。他于 1763 年发表了这方面的论著，对于现代概率论和数理统计的发展都具有很重要的作用。他的许多术语一直被沿用至今，成为了数学发展史上一位举足轻重的人物。

贝叶斯学派是现代统计学中与经典频率学派并列的两大学派之一，贝叶斯数据分析就是先验分布在经过了数据所提供的证据修订之后所形成的经验分布。当时由于贝叶斯方法在理论和应用中还存在很多不完善的地方，因此在很长一段时间并未被普遍接受。后来随着统计决策理论、信息论和经验贝叶斯方法等理论和方法的创立和应用，贝叶斯方法很快显示出他的优点，成为十分活跃的一个方向。

在数据挖掘中，构造分类模型可以使用许多不同的方法，如决策树、贝叶斯方法、神经网络分类法等。通过对分类算法的比较分析，贝叶斯方法可以与决策树算法和神经网络算法相媲美。基于贝叶斯方法的分类器由于以完善的贝叶斯理论为基础，有较强的模型表示、学习和推理能力，使得基于贝叶斯方法的分类器的研究和应用成为模式识别、人工智能和数据挖掘等领域的研究热点。对于大型数据的检测，朴素贝叶斯分类法已表现出高准确率与高速度。贝叶斯分类是统计学分类方法，是建立在经典的贝叶斯概率理论基础上的基于统计方法的分类模型，本章主要介绍贝叶斯理论和各种贝斯分类模型。

3.1　贝叶斯分类理论

贝叶斯的理论可以粗略地被简述成一条原则：为了预见未来，必须要看看过去。贝叶斯的理论表示未来某件事情发生的概率可以通过计算它过去发生的频率来估计。贝叶斯从本质上来说，每件事都有不确定性，有不同的概率类型。例如，一名研究人员把塑料图钉往上抛，要看看它钉头朝上落地的概率有多大，或者有多少可能性是侧面着地，而钉子是指向什么方向的，成型过程中的误差，重量分布和其他的因素都会影响该结果。

贝叶斯方法是一种研究不确定性的推理方法。不确定性常用贝叶斯概率表示，它是一种主观概率。通常的经典概率代表事件的物理特性，是不随人意识变化的客观存在。而贝叶斯概率则是人的认识，是个人主观的估计，随个人的主观认识的变化而变化。某一事件的贝叶斯概率只指个人对该事件的置信程度，因此是一种主观概率。一个企业家认为“一

项新产品能够在未来市场上销售”的概率是0.8，这里的0.8是根据他多年的经验和当时的一些市场信息综合而成的个人信念。同样的例子还有，铁路货运的客户群划分为A类客户（关键客户）、B类客户（主要客户）、C类客户（普通客户）三个类别，在货运现场若一个客户被判定为B类客户的概率是0.6，这里的0.6是根据从先前的训练集训练后得出的分类模型和该客户此时的托运信息综合而成的主观概率。

（1）已知事件A发生的条件下，事件B发生的概率，叫做事件B在事件A发生下的条件概率，记为$P(B \mid A)$，其中$P(A)$叫做先验概率，$P(B \mid A)$叫做后验概率，计算条件概率的公式为：

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} \qquad \text{（公式3-1）}$$

条件概率公式通过变形可以得到乘法公式为：

$$P(A \cap B) = P(B \mid A)P(A) \qquad \text{（公式3-2）}$$

（2）设A,B为两个随机事件，如果有$P(AB) = P(A)P(B)$成立，则称事件A和B相互独立。此时有$P(A \mid B) = P(A)$，$P(AB) = P(A)P(B)$成立。

设$A_1,A_2,\cdots,A_n$为n个随机事件，如果对其中任意$m(2 \leqslant m \leqslant n)$个事件$A_{K_1},A_{K_2},\cdots,A_{K_m}$，都有

$$P(A_{K_1},A_{K_2},\cdots,A_{K_m}) = P(A_{K_1})P(A_{K_2})\cdots P(A_{K_m})$$

成立，则称事件$A_1,A_2,\cdots,A_n$相互独立。

（3）设$B_1,B_2,\cdots,B_n$为互不相容事件，$P(B_1) > 0$，$i = 1, 2, \cdots, n$，且$\bigcup_{i=1}^{n} B_i = \Omega$，对任意的事件$A \subset \bigcup_{i=1}^{n} B_1$，计算事件$A$概率的公式为：

$$P(A) = \sum_{i=1}^{n} P(B_i)P(A \mid B_i) \qquad \text{（公式3-3）}$$

公式3-3称为全概率公式。

设$B_1,B_2,\cdots,B_n$为互不相容事件，$P(B_i) > 0$，$i = 1, 2, \cdots, n$，$P(A) > 0$，则在事件A发生的条件下，事件B_i发生的概率为：

$$P(B_i \mid A) = \frac{P(B_iA)}{P(A)} = \frac{P(B_i)P(A \mid B_i)}{\sum_{i=1}^{n} P(B_i)P(A \mid B_i)} \qquad \text{（公式3-4）}$$

称该公式3-4为贝叶斯公式，即贝叶斯定理。

贝叶斯定理给出了一种根据新的数据不断更新后验分布的方法，它将事件的先验概率和后验概率巧妙地联系起来，利用先验信息和样本数据信息确定事件的后验概率。

令$U = \{A_1,A_2,\cdots,A_n,C\}$为随机变量的有限集，其中$A_1,A_2,\cdots,A_n$是属性变量，类变量$C$的取值范围为$\{c_1,c_2,\cdots,c_k\}$，$a_i$是属性$A_i$的取值。实例$a = (a_1,a_2,\cdots,a_n)$属于$c_j(1 \leqslant j \leqslant k)$的概率，可由贝叶斯公式（3-4）表示为：

$$P(c_j \mid a_1,a_2,\cdots,a_n) = P(a_1,a_2,\cdots,a_n \mid c_j)P(c_j)/P(a_1,a_2,\cdots,a_n) \qquad \text{（公式3-5）}$$

其中，$P(c_j)$是类c_j的先验概率，$P(c_j \mid a_1,a_2,\cdots,a_n)$是$c_j$类的后验概率。先验概率独立于新样本数据，而后验概率则反映了新样本数据对类c_j的影响。

由此我们可以看出，所谓的先验和后验是相对而言的。当我们需要利用新数据更新参数的分布密度时，已知的参数分布密度就是先验分布密度，更新后的参数分布密度是后验

分布密度。这一更新过程可以反复进行，只要有新的数据信息，我们就可以根据贝叶斯定理对先验分布密度进行更新。

一般来说，先验分布反映人们在样本数据获得之前对参数的认识，后验分布则是反映人们在获得样本数据之后对参数的认识。两者之间的差异是由于样本数据出现后对参数的一种调整。例如，在对货票库中的货运实例进行分类时，训练集中 A 类客户数目所占的比例为 27%，B 类客户为 23%，C 类客户为 50%，也就是说，若对某一客户的托运信息一无所知，我们就会认为该客户属于 A 类客户的概率是 0.27，属于 B 类客户的概率是 0.23，属于 C 类客户的概率是 0.5，这就是先验概率。当对该客户的托运信息全面了解后，利用贝叶斯分类模型并结合该客户的托运信息，通过查询各属性的条件概率表，然后将相应的各条件概率相乘，最后将相乘结果化成百分比，这样可将该客户属于每类客户的概率调整为：属于 A 类客户的概率是 0.70，属于 B 类客户的概率是 0.26，属于 C 类客户的概率是 0.04，这就是后验概率。

在对货票库中的数据进行预处理并做聚类分析后，形成的训练集中的随机变量集 U = {计费里程，总标重，运费，运费收入率，客户类别}，其中“计费里程，总标重，运费，运费收入率”是属性变量，类变量“客户类别”的取值范围是 $\{A,B,C\}$。若一个新货主的托运信息为 a = (2655，124，9840.4，7411.6)，由贝叶斯公式 (3-4)，该货主属于类别 A、B、C 的概率可表示为：

$P(A \mid 2655,124,9840.4,7411.6) = P(2655,124,9840.4,7411.6 \mid A)P(A)/P(2655,124,9840.4,7411.6)$

$P(B \mid 2655,124,9840.4,7411.6) = P(2655,124,9840.4,7411.6 \mid B)P(B)/P(2655,124,9840.4,7411.6)$

$P(C \mid 2655,124,9840.4,7411.6) = P(2655,124,9840.4,7411.6 \mid C)P(C)/P(2655,124,9840.4,7411.6)$

根据概率的链规则，公式 3-5 可以表示为：

$$P(c_j \mid a_1,a_2,\cdots,a_n) = a \cdot P(c_j) \cdot \prod_{i=1}^{n} P(a_i \mid a_1,a_2,\cdots,a_{i-1},c_j) \quad \text{(公式 3-6)}$$

其中 $a = 1/P(a_1,a_2,\cdots,a_n)$。

3.2 贝叶斯分类模型

朴素贝叶斯分类 (Naive Bayes Classifier，NBC) 是目前公认的一种简单而有效的概率分类方法，其性能可与决策树、神经网络等算法相竞争，在某些领域中甚至表现出更优的性能。朴素贝叶斯分类假定了一个属性值对给定类的影响独立于其他属性的值，这一假定称作类条件独立。

贝叶斯分类技术就是利用基于贝叶斯公式衍生出来的相关技术来对数据进行分类的方法，通常把这种基于贝叶斯方法构造的分类模型称为贝叶斯分类器。如上所述，贝叶斯公式是一种从先验概率推算后验概率的公式，贝叶斯分类技术的运用也多见于对未知可能性的预测，这种预测是通过事件发生的概率来反映可能性的大小。贝叶斯分类器的分类原理是首先获得某对象的先验概率，然后利用贝叶斯公式计算出其后验概率，即该对象属于某

一类的概率，选择具有最大后验概率的类作为该对象所属的类。先验概率可以通过对已有数据归纳学习或专家经验等方式获得。

3.2.1 贝叶斯分类的特点主要包括：

（1）能够进行增量学习（如图 3－1 所示）。当有新增样本时，无须重新开始学习，而是将原来的数据作为先验知识，新增样本作为样本信息进行学习。

（2）能够结合先验知识和样本数据。贝叶斯分类综合先验信息和样本信息，既可避免只使用先验信息可能带来的主观偏见和缺乏样本信息时的盲目搜索与计算，也可避免只使用样本信息可能产生的噪音影响。因此，适用于具有概率统计特征的数据挖掘和知识发现问题，尤其是样本难以获得或代价较高的领域。

（3）具有不确定性知识表达能力。贝叶斯分类可以给出待分类样本属于各个类别的概率，将样本标记为具有最大概率值的类别。

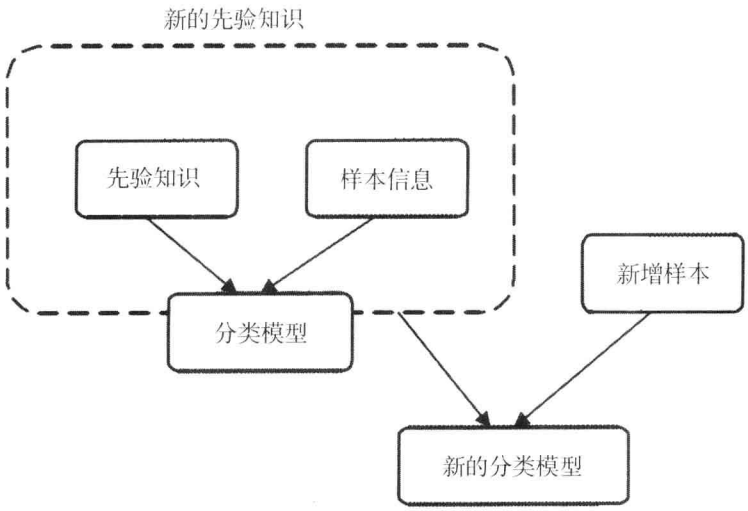


图 3－1　增量学习

根据给定的训练集归纳出分类器是数据挖掘的一项重要而基本的任务。在众多的分类器中（决策树、决策表、神经网络和粗糙集分类器等），朴素贝叶斯分类器以简单的结构和良好的性能受到人们的关注，在理论上它在满足其限定条件下是最优的，针对其较强的限定条件，可以尝试着减弱独立条件以扩大最优范围，从而产生更好的分类器。

3.2.2 朴素贝叶斯分类器

贝叶斯分类器的分类原理是首先获得某对象的先验概率，然后利用贝叶斯公式计算出其后验概率，即该对象属于某一类的概率，选择具有最大后验概率的类作为该对象所属的类。先验概率可以通过对已有数据归纳学习或专家经验等方式获得。

贝叶斯分类器从理论上可以证明是最优分类器，但由于需要直接进行联合概率计算，其效率会比较低，并且容易造成对例子的过度拟合。因此虽然贝叶斯分类器在理论上是最优的分类器，但由于分类器的学习需要完整的样本集才能获得对全部情况的概率估计，并且在进行参数估计时需要极其复杂的计算，因此贝叶斯分类器也是最复杂的一种概率分类

器，在实际应用中较少直接使用。

在贝叶斯分类器诸多算法中，朴素贝叶斯分类模型是最早的。它的算法逻辑简单，构造的朴素贝叶斯分类模型结构也比较简单，运算速度比同类算法快很多，分类所需的时间也比较短，并且大多数情况下其分类精度也比较高，因而在实际中得到了广泛的应用。该分类器有一个朴素的假定：以属性的类条件独立性假设为前提，即在给定类别状态条件下，属性之间是相互独立的。朴素贝叶斯分类器的结构示意图如图 3 - 2 所示。

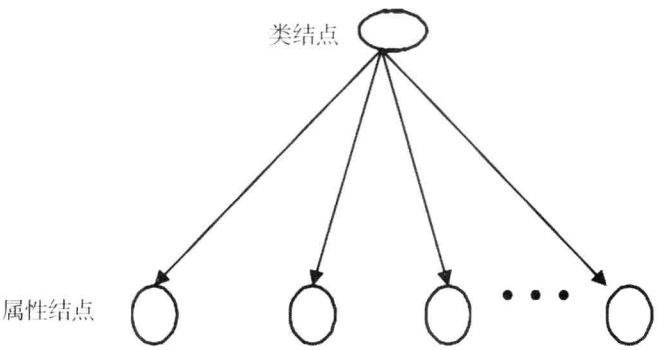


图 3 - 2　朴素贝叶斯分类器示意图

朴素贝叶斯分类模型假定特征向量的各分量间相对于决策变量是相互独立的，也就是各分量独立作用于决策变量。尽管这一假定在一定程度上限制了朴素贝叶斯分类模型的适用范围，但在实际应用中，大大降低了贝叶斯网络构建的复杂性。朴素贝叶斯分类模型已成功地应用到聚类、分类等数据挖掘的任务中。

3.2.3　朴素贝叶斯的算法过程

（1）每个数据样本用 n 维特征向量 $X = \{x_1, x_2, \cdots, x_n\}$ 来表示，分别描述对 n 个属性 $A_1, A_2, \cdots, A_n$ 样本的 n 个度量。

（2）假定有 m 个类 $C_1, C_2, \cdots, C_m$。给定一个未知的样本数据 X（即没有类标号），分类法将预测 X 属于具有最高后验概率（条件 X 下）的类 C_i。朴素贝叶斯分类将未知的样本分配给类，当且仅当：

$$P(C_i \mid X) > P(C_j \mid X), 1 \leqslant i, j \leqslant m, j \neq t$$

这样，最大化 $P(C_i \mid X), P(C_i \mid X)$ 最大的类 C_i 称为最大后验假设，根据贝叶斯定理：

$$P(C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{P(X)}$$

（3）由于 $P(X)$ 对于所有的类为常数，只需要 $P(X \mid C_i)P(C_i)$ 达到最大即可；如果类的先验概率 $P(C_1) = P(C_2) = \cdots = P(C_m) = \dfrac{1}{m}$，$\max P(C_i \mid X)$ 等价于 $\max P(X \mid C_i)$。一般来说，类的先验概率可以用 s_i/s 去估计概率 $P(C_i)$，其中 s_i 是类 C_i 的训练样本数，s 是训练样本总数。

（4）给定具有许多属性的数据集，计算 $P(X \mid C_i)$ 的过程可能相当复杂。为降低计算 $P(X \mid C_i)$ 的复杂性，可以做类条件独立的朴素假定。对于给定样本的类标号，假定属性值相互条件独立，即：

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

概率 $P(x_1 \mid C_i), P(x_2 \mid C_i), \cdots, P(x_n \mid C_i)$ 可以由训练样本估计，其中：

①如果 A_k 是离散属性，则 $P(x_k \mid C_i) = s_{ik}/s_i$，其中 s_{ik} 是属性 A_k 上具有值 x_k 的类 C_i 的训练样本数，而 s_i 是 C_i 中的训练样本数。

②如果 A_k 是连续值属性，则通常假定该属性服从高斯分布。因而，

$$P(x_k \mid C_i) = g(x_k, u_c, \sigma_{c_i}) = \frac{1}{\sqrt{2\pi}\sigma_{c_i}} e^{\frac{(x_k - u_{c_i})^2}{2\sigma_{c_i}^2}}$$

其中给定类 C_i 的训练样本属性 A_k 的值，$g(x_k, u_{c_i}, \sigma_{c_i})$ 是属性 A_k 的高斯密度函数，而 u_{c_i}, σ_{c_i} 分别为平均值和标准差。

(5) 为对位置样本 X 分类，计算 $P(X \mid C_i)P(C_i)$。样本 X 被指派到类别 C_i 中，当且仅当

$$P(C_i \mid X) > P(C_j \mid X), 1 \leqslant i, j \leqslant m, j \neq i$$

换言之，X 被指派到其 $P(X \mid C_i)P(C_i)$ 最大的类别 C_i 中。

3.2.4 朴素贝叶斯分类器的优缺点

朴素贝叶斯分类算法的优点：

(1) 算法逻辑简单，且易于实现；

(2) 算法实施的时间和空间开销小；

(3) 算法性能稳定，对于不同特点的数据其分类性能差别不大，即模型的健壮性比较好。

尽管在实际情况中难以满足朴素贝叶斯模型的属性类条件独立性假定，但它的分类预测效果在大多数情况下仍比较精确。原因有如下几个：要估计的参数比较少，从而加强了估计的稳定性；虽然概率估计是有偏的，但人们大多关心的不是它的绝对值，而是它的排列次序，因此有偏的概率估计在某些情况下可能并不要紧；现实中很多时候已经对数据进行了预处理，比如对变量进行了筛选，可能已经去掉了高度相关的量等。除了分类性能很好外，贝叶斯分类模型还具有形式简单、可扩展性很强和可理解性很好等优点。

朴素贝叶斯分类器的缺点是属性间类条件独立的这个假定，而很多实际问题中这个独立性假设并不成立，如果在属性间存在相关性的实际问题中忽视这一点，会导致分类效果下降。

虽然在某些不满足独立性假设的情况下，朴素贝叶斯分类模型的分类效果仍比较好，但是大量研究表明可以通过各种改进方法来提高朴素贝叶斯分类器的性能。朴素贝叶斯分类改进方法主要有两类：一类是弱化属性的类条件独立性假设，在朴素贝叶斯分类器的基础上构建属性间的相关性，如构建相关性度量公式，增加属性间可能存在的依赖关系；另一类是构建新的样本属性集，期望在新的属性集中，属性间存在较好的类条件独立关系。

朴素贝叶斯分类器在文档分类中很常用。其基本思想是利用单词和类之间的联合概率来估计给定文档属于类的概率。朴素贝叶斯方法假设不同单词在给定类别下的条件概率分布是互相独立的。该假设使得朴素贝叶斯分类器不需要计算单词之间的联合分布概率，因此其速度远快于非朴素贝叶斯模（达到指数复杂度）。

3.3 贝叶斯分类模型的改进研究

3.3.1 树扩张型贝叶斯分类模型

（1）TAN 分类模型概述

朴素贝叶斯分类器（NBC）需要一个很强的属性之间条件独立性的假设，而这些假设在许多实际问题中并不成立，如果忽视这一点会引起分类的误差。由于在 NBC 中，所有的属性结点都属于类结点的马尔可夫覆盖，所以可以对 NBC 进行增强，保留其结构特点，放松它的独立性假设，使属性变量之间存在一定的简单依赖关系，称这类分类器为树扩展的 TAN 贝叶斯网络分类器（Tree Augmented Naive Bayesian Classifier）。除类结点为父结点外，TAN 贝叶斯网络要求属性结点最多只能有一个属性父结点，如图 3-3 所示。

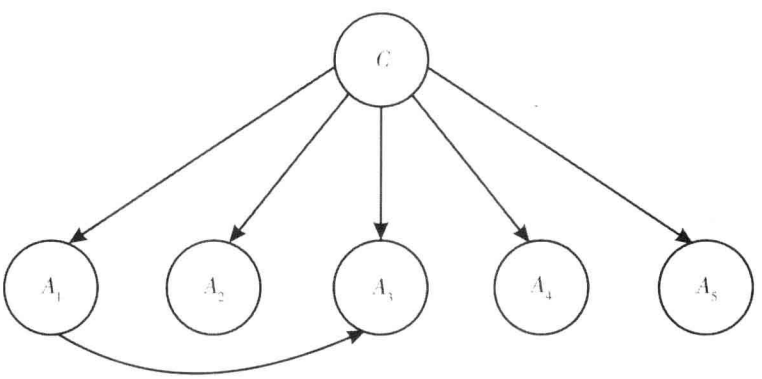


图 3-3　树扩展的 TAN 贝叶斯分类器示意图

设 $X = \{X_1, X_2, \cdots, X_n, C\}$，其中 $\{X_1, X_2, \cdots, X_n\}$ 是属性变量集，C 表示类节点 $\prod_{x_i}$，用表示属性节点 X_i 的父节点集合，由于在 TAN 树形结构中，类节点是树的根节点，没有父节点，属性节点除了类节点以外最多只有一个父节点，所以有：$C \subset \prod_{x_i}$，且 $\prod_C = \phi$。因此，函数 π 也就定义了 TAN 贝叶斯网络，给定属性节点之间的条件互信息函数：

$$I_p(X_i; X_j \mid Z) = \sum_{x,y,z} P(x_i, x_j, z) \log \frac{P(x_i, x_j \mid z)}{P(x_i \mid z) P(x_j \mid z)}$$

具有最大似然的 TAN 贝叶斯网络的构造如下：

① 通过训练集计算属性对之间的条件互信息 $I_{P_D}(A_i; A_j \mid C)$；

② 简历一个以 $I_{P_D}(A_i; A_j \mid C)$ 为弧的权重的加权完全无向图；

③ 简历一个最大权重跨度树；

④ 选择一个根节点，设置所有边的方向是由根节点向外，把无向树转换为有项树；

⑤ 增加一个类变量节点及类变量节点与属性节点之间的弧。

建立最大权重跨度树的方法是：首先把边按权重由大到小排序，之后遵照选择的边不能构成回路的原则，按照边的权重由大到小的顺序选择边，这样由所选择的边构成的树便是最大权重跨度树。贝叶斯网络分类器 TAN 是对朴素贝叶斯网络分类器进行有效改进的

分类器，它既有朴素贝叶斯分类器的简单性，又有比朴素贝叶斯分类器更好的分类性能，当然，还可以对贝叶斯网络分类器 TAN 进行有效的改进。

（2）TAN 分类模型的特点

与朴素贝叶斯分类器、一般贝叶斯网络分类器相比较，TAN 有如下优点：

①TAN 放松了朴素贝叶斯分类器严格的条件独立性假设，并借鉴贝叶斯网络中表示依赖关系的方法，扩展了朴素贝叶斯的结构，使其能容纳属性间存在的依赖关系，但对其表示依赖关系的能力加以限制。由于综合了朴素贝叶斯的简易性和贝叶斯网络表示依赖关系的能力，TAN 体现了学习的效率与准确地描述属性之间相关性的一种适当的折中，是具有较好的综合性能的分类模型。与一般贝叶斯网络相比，TAN 结构的学习时间明显降低；与朴素贝叶斯网络相比，其分类性能明显提高。

②TAN 克服了 K2 算法由于缺乏先验知识而无法确定或只能盲目确定节点次序的问题，网络结构更加合理。

例如，可能会有这样的情况：待分类样本的属性值 A_i 和 A_j 对分类的影响都不大，即 $P(a_i \mid c)$ 和 $P(a_j \mid c)$ 的值比较低，但从节点 A_i 到 A_j 有一条扩展边，而 $P(a_i \mid a_i, c)$ 的值却很高，这时，朴素贝叶斯分类器就会过度降低该样本属于类别 c 的概率，而 TAN 分类器却可以避免这一点。但 TAN 要求所有属性节点必须构建成树形结构，在简化了一般贝叶斯网络的同时，也体现出它的不合理之处。另外，在文本分类领域，由于文本数据的属性个数很多并且属性之间存在某些依赖关系，用 TAN 构建文本分类模型，添加的扩展弧较多，结构模型比较复杂，计算复杂度和空间复杂度都是属性个数的二次项级，算法所需的存储空间很大，甚至出现使系统无法正常运行的情况。

3.3.2 基于属性间的关系

（1）属性分组技术

Kononcnko 提出一种采用穷尽搜索的属性分组技术，假定同个组内的属性之间可能是相互依赖的，但是组与组之间是满足独立性假设的属性集合。也就是说，独立性假设弱化为这些属性组之间的独立性。但是，这种算法的复杂性远远高于朴素贝叶斯分类器，而且在现实世界中，属性可以完全被分成独立的子集合只是少数情况。

（2）属性删除技术

适用于存在冗余属性的情况。Langley 和 Sage 提出了一种基于属性删除的选择性贝叶斯分类器。当存在一些属性依赖于其他属性，特别是存在冗余属性时，属性删除方法确实能够改善朴素贝叶斯分类器的预测精确度。

（3）局部朴素贝叶斯分类器

适用于属性之间依赖的情形比较复杂的情况。这种方法是为属性变量的每种取值（或某个范围）建立一个朴素贝叶斯分类器。也就是说，单一的全局朴素贝叶斯分类器被许多局部朴素贝叶斯分类器所代替，将属性独立性假设放宽到只要局部属性独立就可以了。Kohavi 将朴素贝叶斯分类器和决策树相结合，用一棵决策树来分割实例空间，在每个叶子结点上建立局部朴素贝叶斯分类器。

Zhang 和 Webb 等利用懒惰式学习策略提出了一种懒惰式贝叶斯规则（Lazy Bayasian Rule，LBR）和学习技术，该方法将懒惰式技术应用到局部朴素贝叶斯规则的归纳中。该算法虽然较大地提高了分类精确度，但是效率很低。

为了提高 LBR 的效率，Wang 和 Webb 给出了一种启发式 LBR 算法 HLBR，可以有效地提高学习效率。LBR 和 HLBR 是目前该方向上的最新研究成果之一。

（4）构造新属性或概率调整技术

该技术适用于某些属性依赖于其他属性的情形。Pazzani 等提出了通过相互依赖的属性构造一个新属性，并用新属性取代原来相互依赖的那些属性的方法，这种方法也可以视作事件的条件概率调整技术。

Wang 和 Webb 等提出了一种准懒惰式（Semi - Lazy）的限制性贝叶斯网络分类器的条件概率调整方法，在某些情况下可以减小误分类率。

（5）建立依赖关系技术

该技术适用于属性之间相互依赖的情形比较复杂的情况。

Friedman 等提出了一种树扩张型贝叶斯方法（Tree - Augmented Baycsian Classifier，TAN）（即 3.3.1 讨论的内容）。这种方法的基本思路是放宽朴素贝叶斯的独立性假设条件，扩展朴素贝叶斯的结构，使其能够容纳属性间存在具有某种特征的依赖关系。Friedman 利用条件相互信息（Conditional Mutual Information）建立属性之间的依赖关系矩阵，构造一棵最大权重生成树作为一个分类器。由于 TAN 限制每个属性结点最多有一个非类变量（类标）的父结点，也就是说每个属性结点最多仅依赖于一个非类标结点，使其表示依赖关系的能力受到限制。

石洪波等提出了一种限定型的双层贝叶斯分类模型 DLBAN（Double - Lever Baycsian Network Augmented Naive Bayes）。这种方法的出发点是通过对属性空间的搜索，找出一些对其他属性有较强影响的属性，那么所有其他的属性仅通过与这些属性的关联就可以将重要的依赖关系表示出来。

Zhang 和 Link 提出了种“树桩网络（Stump Network）”的模型。该模型的灵感来自于现实世界中如果数据集中含有大量的属性，则这些属性倾向于聚集成组，为了反映这种依赖，该算法先将所有属性构造成一个个简单的组，即“树桩”，然后再在树桩之间构造连接。该模型适合于含有大量属性的数据集。

3.3.3 整体分类技术

（1）懒惰式（Lazy）学习策略相对于急切式（Eager）学习策略的不同之处在于，前者是在分类时间，对于每个来分类的实例，临时建立起一个分类器来进行分类；后者是在训练时间就先行建立了分类器，在分类时间上分类所有要分类的实例。Friedman 等的 TAN 是采用急切式学习策略的分类器，即对于给定的训练实例集合，在训练阶段就建立好一个树扩张型贝叶斯网络分类器，在分类阶段直接地使用该分类器来分类每个待分类实例。而 LBR 采取了一种完全懒惰式的学习方法，即在训练阶段仅仅产生所需要的频率表和朴素贝叶斯分类器的误分类实例数目，而在分类阶段，对每个待分类实例建立其局部的贝叶斯网络分类器。一般而言，对于同种模型技术，懒惰式学习策略在分类精度上优于急切式学习策略，但是在学习效率上却大大劣于急切式学习策略。

（2）显现模式技术

显现模式技术（Emerging Patterns）是指在数据集合中表现活跃的部分属性值组合序列，将显现模式技术到分类器的构造中是一种新的研究方法。

3.4 贝叶斯网络

3.4.1 贝叶斯网络的定义

朴素贝叶斯分类器基于条件独立假设，即在给定类别的条件下，各属性变量相互独立。

这一假定使得朴素贝叶斯分类器具有较为简洁的结构。但实际生活和应用中，变量之间可能存在依赖关系。贝叶斯信念网络是对朴素贝叶斯分类器的扩展，它放宽了条件独立假设，允许存在依赖关系的变量间进行加边。贝叶斯网络结构如图 3 -4 所示。

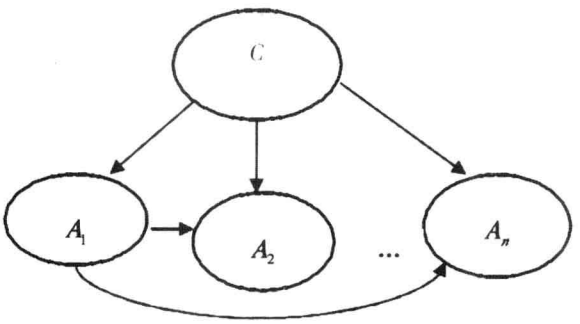


图 3 -4　贝叶斯网络示意图

贝叶斯网络定义如下：

有向无环图 G 可以用（V，E）表示，V 是节点的集合，E 是有向边的集合。从节点 X 到节点 Y 的有向边用（X，Y）表示，X 和 Y 是相邻节点，且 X 是 Y 的父节点，Y 是 X 的子节点。没有父节点的节点成为根节点。

一个贝叶斯网络用图形表示一组随机变量之间的概率关系，它由两个部分构成：

（1）具有 n +1 个节点的有向无环图 G。图中的每个节点代表一个属性，节点间的有向边代表了属性间的依赖性。

（2）条件概率表 CPT。每一个有父节点的节点都有一个条件概率表，定量地表达了该节点同其父节点的依赖关系。

如图 3 -5 所示，描述了一个简单的贝叶斯网络结构。其中，C 是根节点，它由两个子节点，分别为 A_1 和 A_2 是节点 A_4 的父节点。节点 A_3 的父节点是 A_1。

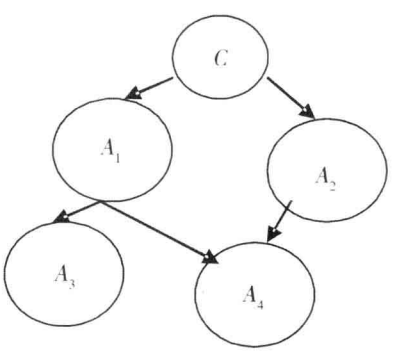


图 3 -5　一个简单的贝叶斯网络示意图

贝叶斯网络具有条件独立的特性：

贝叶斯网络中的一个节点，若它的父节点已知，则它条件独立于它的所有非后代节点。依据贝叶斯网络的定义，每个节点的条件概率表描述如下：

（1）如果节点 X 没有父节点，则表中只包含先验概率 $P(X)$。

（2）如果节点 X 只有一个父节点 Y，则表中包含条件概率 $P(X \mid Y)$。

（3）如果节点 X 有多个父节点 $\{Y_1, Y_2, \cdots, Y_k\}$，则表中包含条件概率 $P(X \mid Y_1, Y_2, \cdots, Y_k)$。

贝叶斯网络的学习包括结构学习和参数学习。结构学习通过对训练集的学习，寻找与其最为匹配的网络结构，常用方法包括基于评分的方法和基于条件独立性的方法。参数学习根据数据集的完备性可以分为不完备数据的参数学习和完备数据的参数学习，不完备数据的参数学习一般采用近似方法，如 EM 算法等，完备数据的参数学习一般指定概率分布簇，然后对指定的概率分布簇估计参数。

3.4.2 贝叶斯网络特点

贝叶斯网络自十几年前提出以来，随着理论研究的不断深入和应用水平的不断提高，已经得到学术界更加广泛地认可和重视。对于复杂问题的决策、不确定性知识的推理，以及智能诊断和可靠性评估等领域，贝叶斯网络更加体现出其明显的优势和魅力。任何一种方法技术都具有其自身的特点，贝叶斯网络与决策树、人工神经网络、遗传算法等其他方法相比，主要有以下几个特点：

（1）强大的知识表达和推理能力

贝叶斯网络可以有效地表达和分析不确定性问题，实现定性分析与定量分析的有机结合，运用网络结构进行定性表达，运用概率参数进行定量描述。它既能够从原因推断结果（预测），也能够从结果找出原因（诊断）。同时，还可以实现对含有噪声数据、缺少某一数据变量或不完备数据集的表达和推理，因此，贝叶斯网络已发展成为处理不确定性知识的一种有力工具。特别是在费用昂贵的实验中或缺乏样本数据的情况下，对于决策和预测具有更加重要的意义。

（2）灵活的学习机制

通过贝叶斯网络的学习，可以发现潜在有用的模式或关系，实现对数据实例的分类、聚类和预测。目前，已经成为知识发现与数据挖掘中的一种主要方法。

贝叶斯网络随着专家知识和数据样本的更新，其结构和参数都将得到相应地修正与更新，形成一种高效灵活的学习机制，类似于人类的学习方式和认知过程。

（3）开放的决策体系

贝叶斯网络有机地结合先验信息和样本数据，将主观和客观有机地结合起来，更加全面客观地反映数据对象内在的联系与本质。不仅可以避免由于主观因素可能造成的偏见，而且可以避免样本数据带来的噪音问题。其次，贝叶斯方法将静态和动态结合起来，充分利用前人的知识和经验，这符合认识的一般规律，对于大部分自然现象，在不同的时间可以得到完全相同的实验结果，于是静态概率可以对动态过程有广泛的适用性。然而，事物总是在发展和变化的，在进行决策时，必须将历史的经验和变化的情况结合起来。先验概率是相对静态的经验知识，样本信息反映动态的变化情况，得到的后验概率则是两者结合的产物。贝叶斯网络运用图形的方式描述数据或变量之间的依赖关系和因果关系。它是对

数据对象的高度抽象与概括，将变量之间潜在的关联性用简洁的图解模型表达出来，表达的语义清晰、自然直观。由于贝叶斯网络运用严密的数学推理方法，推理得出的结果和结论可信度强，也便于解释和易于理解。

3.5 贝叶斯网络应用实例

表3-1给出了对铁路货运客户进行类别划分后的部分历史货票记录。其中，前4个属性在原数据库中均为数值型属性，但由于贝叶斯网络分类器只能处理离散型属性，因此表中的数据都经过了离散化处理。

表3-1 铁路货运客户分类训练样本集

样本序号	计费里程	总标重	运费	运费收入率	客户类别
1	[-inf -875)	[100 -300)	[1000 -inf)	[4500 -8000)	C
2	[2283 -inf)	[100 -300)	[10 000 -inf)	[4500 -8000)	A
3	[875 -2283)	[100 -300)	[5000 -10 000)	[4500 -8000)	B
4	[-inf -875)	[300 -inf)	[10 000 -inf)	[800 -inf)	C
5	[875 -2283)	[-inf -100)	[5000 -10 000)	[-inf -4500)	B
6	[2283 -inf)	[-inf -100)	[10 000 -inf)	[8000 -inf)	A
7	[-inf -875)	[100 -300)	[5000 -10 000)	[-inf -4500)	C
8	[-inf -875)	[100 -300)	[10 000 -inf)	[8000 -inf)	C
9	[875 -2283)	[-inf -100)	[-inf -5000)	[-inf -4500)	B
10	[-inf -875)	[100 -300)	[10 000 -inf)	[-inf -4500)	C
…	…	…	…	…	…

表3-1中离散化后的每个属性分别划分成三个区间。贝叶斯网络分类器结构的学习首先初始化为朴素贝叶斯网络结构，并且采用K2算法，节点次序按照训练集中的顺序，评分函数使用贝叶斯法，将最大节点数设为2。该训练集经过学习后得出的贝叶斯网络结构如图3-6所示。

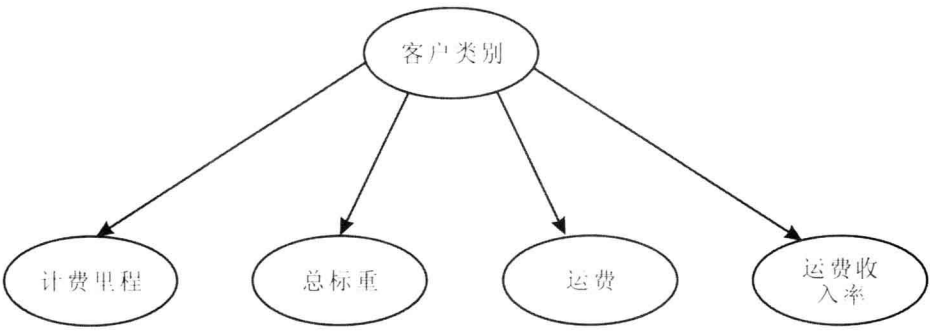


图 3-6　货运客户分类贝叶斯网络结构示意图

类节点的先验概率表和个属性节点的条件概率表分别如表 3-2～表 3-6 所示。

表 3-2　“客户类别”的先验概率表

客户类别		
A	B	C
0.25	0.25	0.5

表 3-3　“计费里程”的条件概率表

	计费里程		
客户类别	[-inf-875)	[875-2283)	[2283-inf)
A	0.077	0.231	0.692
B	0.077	0.846	0.077
C	0.193	0.043	0.043

表 3-4　“总标重”的条件概率表

		总标重		
客户类别	计费里程	[-inf-100)	[100-300)	[300-if)
A	[-inf-875)	0.333	0.333	0.333
A	[875-2283)	0.2	0.6	0.2
A	[2283-inf)	0.455	0.273	0.273
B	[-inf-875)	0.333	0.333	0.333
B	[875-2283)	0.538	0.385	0.077
B	[2283-inf)	0.333	0.333	0.333
C	[-inf-875)	0.304	0.478	0.217
C	[875-2283)	0.333	0.333	0.333
C	[2283-inf)	0.333	0.333	0.333

表 3-5 “运费”的条件概率表

运费				
客户类别	总标重	[-inf -5000)	[5000 -10 000)	[10 000 -inf)
A	[-inf -100)	0.143	0.429	0.429
A	[100 -300)	0.143	0.143	0.714
A	[300 -inf)	0.2	0.2	0.6
B	[-inf -100)	0.333	0.556	0.111
B	[100 -300)	0.143	0.429	0.429
B	[300 -inf)	0.333	0.333	0.333
C	[-inf -100)	0.556	0.333	0.111
C	[100 -300)	0.231	0.231	0.538
C	[300 -inf)	0.143	0.143	0.714

表 3-6 “运费收入率”的条件概率表

运费收入率				
客户类别	运费	[-inf -4500)	[4500 -8000)	[8000 -inf)
A	[-inf -5000)	0.333	0.333	0.333
A	[5000 -10 000)	0.2	0.6	0.2
A	[10 000 -inf)	0.091	0.273	0.636
B	[-inf -5000)	0.6	0.2	0.2
B	[5000 -10 000)	0.333	0.556	0.111
B	[10 000 -inf)	0.2	0.2	0.6
C	[-inf -5000)	0.778	0.111	0.111
C	[5000 -10 000)	0.714	0.143	0.143
C	[10 000 -inf)	0.231	0.385	0.385

包括类属性“客户类别”在内的 5 个变量都是三值变量，则存储联合分布需要 $3^5 - 1 = 242$ 个参数。若将货票的联合概率分解为子集上的 5 个条件概率的乘积，只需要 2 + 6 + 6 + 18 + 18 = 50 个参数，从而大大降低了参数的存储空间。

若一个新货主的信息为 a = (2655,124,9840.4,7411.6)，经离散化后为([2283 - *inf*),[100 - 300),[5000 - 10 000),[4500 - 8000))。通过查询各个属性的条件概率表，计算出该货主各个类别的概率为：

$$P(A \mid a) = P(A)\prod_{i=1}^{4} P(a_1 \mid pa(A_i)) = 0.25 \times 0.692 \times 0.273 \times 0.143 \times 0.6 = 0.00405225$$

$$P(B \mid a) = P(B)\prod_{i=1}^{4} P(a_1 \mid pa(A_i)) = 0.25 \times 0.077 \times 0.333 \times 0.429 \times 0.556 = 0.00152900$$

$$P(C \mid a) = P(C)\prod_{i=1}^{4} P(a_1 \mid pa(A_i)) = 0.5 \times 0.043 \times 0.333 \times 0.231 \times 0.143 = 0.00023650$$

化为百分比：

$P(A \mid a) = 0.00405225/(0.00405225 + 0.00152900 + 0.00023650) = 69.6\%$

$P(B \mid a) = 0.00152900/(0.00405225 + 0.00152900 + 0.00023650) = 26.3\%$

$P(C \mid a) = 0.00023650/(0.00405225 + 0.00152900 + 0.00023650) = 4.1\%$

依据贝叶斯最大后验准则，该货主归属的类别为 A。

第4章　人工神经网络算法

4.1　基本概念及特征

神经网络理论是巨量信息并行处理和大规模并行计算的基础，神经网络不仅是高度非线性动力学系统，而且也是自适应组织系统，可用来描述认知、决策及控制的智能行为。它的中心问题是智能的认知和模拟，从解剖学和生理学来看，人脑是一个复杂的并行系统，它不同于传统的 Neumann 式计算机，更重要的是它具有“认知”、“意识”和“感情”等高级脑功能。以人工方法模拟这些功能，毫无疑问，有助于加深对思维和智能的认识。20 世纪 80 年代初，神经网络的崛起，已对认知和智力本质的基础研究乃至计算机产业都产生了空前的刺激和极大的推动作用。

人工神经网络（Artificial Neural Network，ANN）简称神经网络，它采用物理上可实现的器件或采用计算机来模拟生物体中神经网络的某些结构和功能，用大量的简单计算单元（即神经元）构成的非线性系统，它在一定程度和层次上模仿了人脑神经系统的信息处理、存储及检索功能，因而具有学习、记忆和计算等智能处理功能，在工程领域得到应用。

人工神经网络是模仿脑细胞结构和功能、脑神经结构以及思维处理问题等脑功能的新型信息处理系统。人工神经元网络结构和工作机理基本上是以人脑的组织结构（大脑神经元网络）和活动规律为背景的，它反映了人脑的某些基本特征，但并不是要对人脑部分的真实再现，可以说它是某种抽象、简化或模仿。神经网络的基本组成单元是神经元，在数学上的神经元模型是和在生物学上的神经细胞对应的。或者说，人工神经网络理论是用神经元这种抽象的数学模型来描述客观世界的生物细胞的。生物的神经细胞是神经网络理论诞生和形成的物质基础和源泉。神经元的数学描述必须以生物神经细胞的客观行为特性为依据。因此，了解生物神经细胞的行为特性对人工神经网络研究有着重要的启示。

虽然人工神经网络与真正的生物神经网络有差别，但由于汲取了生物神经网络的部分优点，因此具有一些固有的特性：

（1）自学习、自组织和自适应性。学习和适应要求在时间过程中其系统内部结构和联系方式有改变，神经网络是一种变结构系统，恰好能完成对环境的适应和对外界事物的学习能力。神经元之间的连接多种多样，神经元之间的连接强度具有一定的可塑性，相当于突触传递信息能力的变化，这样，网络可以通过学习和训练进行自组织以适应不同信息处理要求。

（2）人工神经网络是大量神经元的集体行为，并不是各单元行为的简单相加，而表现

出一般复杂非线性动态系统的特性，如不可预测性、不可逆性、有各种类型的吸引子（信息正是“存储”在定点吸引子上）和出现混沌现象等。

（3）大规模并行处理。神经网络在结构上是并行的，而且网络的各个单元可以同时进行类似的处理过程，因此，网络中信息处理是在大量单元中平行而又有层次的进行，运算速度高，大大超过传统的序列式运算的数字机。

（4）分布存储与容错性。一个信息不是存储在一个地方，而是按内容而分布在整个网络上，网络某一处不是只存储一个外部信息，而每个神经元存储多种信息的部分内容。网络的每部分对信息的存储有等势作用。这种分布存储方法是存储区与运算区合为一体的。在神经网络中，要获得存储的知识则采用“联想”的办法，即当一个神经网络输入一个激励时，它要在已存的知识中寻找与该输入匹配最好的存储知识为其解。当然在信息输出时，也要经过一种处理，而不是直接从记忆中取出。这种存储方式的优点在于若部分信息不完全，就是说或者丢失或者损坏甚至有错误的信息，它仍然能恢复出原来正确的完整的信息，系统仍能运行。这就是网络具有容错性和联想记忆功能，自然呈现出较强的鲁棒性。

（5）神经元可以处理一些环境信息十分复杂、知识背景不清楚和推理规则不明确的问题。例如语音识别、手写体识别、医学诊断以及市场估计等，都是具有复杂非线性和不确定对象的控制。在那里，信源提供的模式丰富多彩，有的互相间存在矛盾，而判定决策原则又无条理可循，通过神经网络学习（按照学习法则），从典型事例中学会处理具体事例，给出比较满意的解答。

4.2 人工神经网络的分类

迄今为止，神经网络已经约有 40 种不同的模型，按照不同的分类原则，可以对神经网络进行不同的分类，通常有以下五类。

（1）按照学习方式分类：监督学习网络和无监督学习网络。

（2）按照信息传递规律分类：前馈神经网络（Feedforward NN）、反馈神经网络（Feedback NN）和自组织神经网络（Self - organizing NN）。

（3）按照网络的性能分类：连续型网络和离散型网络、随机型网络和确定型网络。

（4）按照对生物神经系统的层次模拟分类：神经元层次模型、网络层次模型、组合式模型、神经系统层次模型和智能型模型。

（5）按照突触性质分类：一阶线性并联网络和高阶非线性并联网络。

在人工神经网络的设计与应用过程中，人们较多地考虑神经网络的互联结构，包括 5 种典型结构。

（1）前馈网络。前馈网络中，神经元是分层排布的，每个神经元只与前一层的神经元相连，图 4 - 1（a）的最下一层为输入层，最上一层为输出层，中间层称为隐层，隐层的层数可以是一层或多层。

（2）有反馈的前馈网络。如图 4 - 1（b）所示，网络本身仍然是前馈型的，在输出层和输入层之间存在一个反馈回路。该种神经网络的输入层在接受外界的输入信号时，也接受网络自身的输出反馈信号。输出反馈信号可以是原始输出信号，也可以是经过处理的输

出信号；可以是当前时刻的信号，也可以是经过一段延迟的信号。这种网络特性使之经常用于实时信号处理、系统控制等需要根据系统当前状态进行实时调节的系统。

（3）前馈内层互联网络。如图 4－1（c）所示，在隐层和输出层的同一层内存在互相连接，实现信息交换的同时也可以形成互相制约，而从外部看还是一个前向网络，很多自组织网络大都存在内层互联的结构。

（4）反馈型全互联网络。图 4－1（d）所示的网络是一种单层全互联网络，每个神经元的输出都与其他神经元相连，如 Boltzmann 机和 Hopfield 神经网络都是属于这一类网络。

（5）反馈型局部连接网络。图 4－1（e）所示的网络是一种单层网络，它的每个神经元的输出只与其周围的神经元相互连接，交换彼此信息从而形成反馈的网络，这类网络也可发展为多层的三角形的结构。

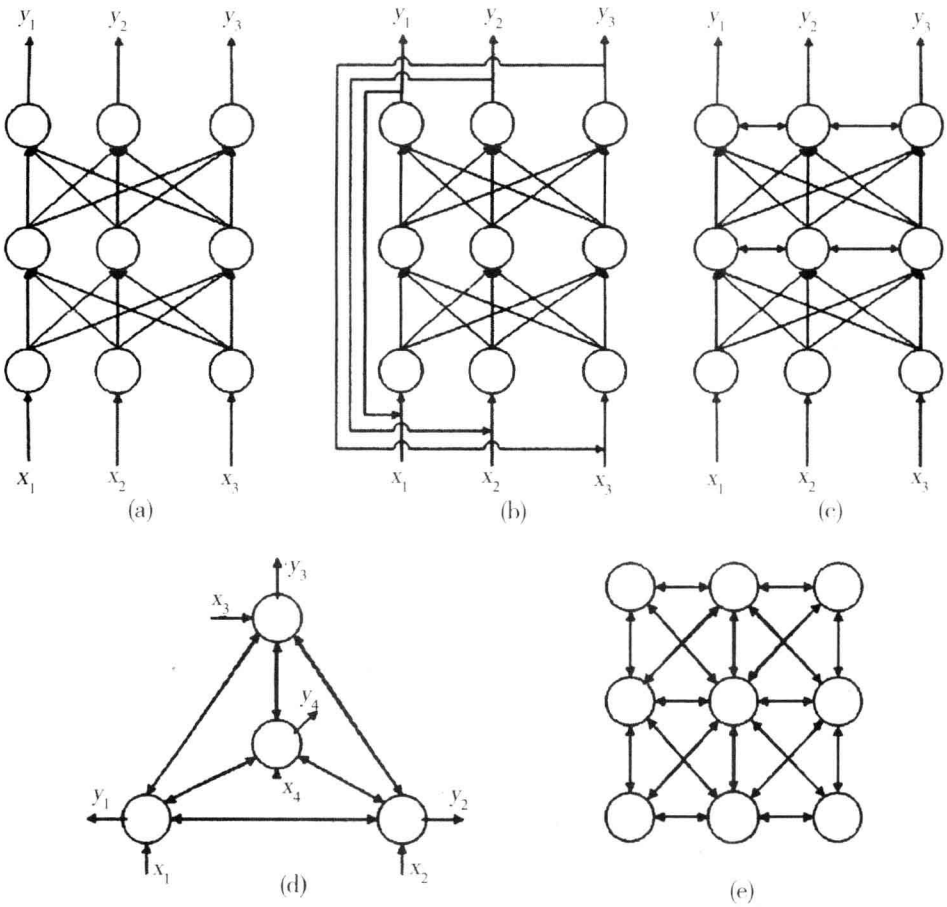


图 4－1　神经网络的拓扑结构

目前，最具代表性的前馈神经网络模型有两种：①BP（Back Propagation，反向传播）神经网络。它是一种多层前馈网络，在多层感知器的基础上增加误差反向传播信号，这样就可以处理非线性的信息，是应用最广泛的网络。②RBF（Radia Basis Function，径向基函数）神经网络。它是在生物局部调节和交叠接受区域知识的基础上提出的一种采用局部接受域来执行函数映射的前馈神经网络。在某些领域的实际应用中，其分析能力、逼近能力和学习速度等方面表现出优于 BP 网络的性能。

4.3 人工神经网络的发展历史

神经网络的理论与应用研究得到了极大的发展，而且已经渗透到几乎所有的工程应用领域。但是，神经网络的发展过程是坎坷的，自 1943 年心理学家 McCulloch 和数学家 Pitts 提出神经元生物学模型（简称 M－P 模型）以来，至今已有 60 多年的历史了。在这 60 多年的发展历史中，大致可以分为以下几个发展阶段。

（1）人工神经网络的兴起阶段

1943 年，W. McCulloch 和 W. Pitts 提出了 MP 模型，从而给出了神经元的最基本模型及相应的工作方式。1949 年，神经生物学家 D. Hebb 发现，脑细胞之间的通路在参与某种活动时将被加强。这一重要规则给出了生理学与心理学间的联系，被称为 Hebb 学习规则。该规则至今还被许多神经网络学习算法所使用。1957 年，F. Rosenblat 提出了感知器模型，这是一个由线性阈值神经元组成的前馈神经网络，可用于分类。1960 年，B. Widrow 和 M. Hoff 提出了自适应线性单元，这是一种连续取值的神经网络，可用于自适应系统。

（2）神经网络的低潮

1969 年，人工智能的创始人 M. Minsky 和 Papert 发表了《感知器》（Perceptrons）一书。在该书中，他们指出：单层 Perceptron 只能作线性划分，多层 Perceptron 不能给出一种学习算法，因此无实用价值。由于 M. Minsky 和 Papert 在人工智能领域的地位，该书在人工神经网络研究人员间引起了极大的反响，因此神经网络研究从此陷入低潮。但是，即便在神经网络研究的低潮时期，也有一些人仍在兢兢业业地研究神经网络，并取得了一些重要成果。其中最著名的是 1982 年由加州理工大学教授 H. Hopfield 提出的 Hopfield 神经网络。在这个用运算放大器搭成的反馈神经网络中，Hopfield 借用 Lyapunov 能量函数的原理，给出了网络的稳定性判据，并为著名的组合优化问题——旅行商问题（TSP）提供了一个新的解决方案。Hopfield 网络可用于联想存储、优化计算等领域。

（3）神经网络的黄金时期

1985 年，Rumelhart 等人给出了多层感知器权值训练的误差反向传播学习算法（BP 算法），从而解决了 Minsky 认为不能解决的多层感知器的学习问题。自此引导了神经网络的复兴，神经网络研究也进入了一个崭新的发展阶段。1982 年，Hopfield 率先提出了一种新的神经网络模型——HNN，并引入了“能量函数”概念，使得网络稳定性研究有了明确的理论依据。同时，HNN 的电子电路物理实验为神经计算机的研究奠定了基础，并将其应用于目前计算机尚难解决的 NP 完全型问题，例如著名的“TSP”问题就取得了很好的效果。从事并行分布处理研究的学者，如 Hinton、Sejnowsky 和 Rumelhart 等，于 1985 年对 HNN 模型引入随机机制，提出了 Boltzmann 机。1986 年，Rumelhart 等人在多层神经网络模型的基础上提出了 BP 算法，解决了多层前向神经网络的学习问题。

近几十年来，许多具备不同信息处理能力的神经网络模型已经被提出来并应用于许多信息处理领域，如模式识别、信号处理、决策辅助、人工智能等方面。神经计算机的研究也为神经网络的理论研究提供了许多有利的条件。各种神经网路模拟软件包、神经网络芯

片以及电子神经计算机的出现，体现了神经网络领域的各项研究均取得长足进展。虽然人们已经对神经网络在人工智能领域的巨大研究潜力达成共识，但是，人类对自身大脑的研究，尤其是对其中智能信息处理机制的了解，还很肤浅。现有的研究成果仅仅处于起步阶段，更需要我们付出不懈的努力。

概括以上简要介绍，可以看出，当前又处于神经网络理论的研究高潮，特别是神经网络在不同领域的应用。不仅给新一代计算机的研究带来巨大影响，而且将推动整个人工智能研究领域的发展。但另一方面，由于问题本身的复杂性，不论是神经网络原理自身，还是正在努力进行探索和研究的神经计算机，目前都还处于发展起步阶段。它的应用影响力和最终所能达到的目标，目前还不十分明确，还有待于继续深入的研究。

4.4 人工神经元模型

4.4.1 生物神经元

人工神经网络是对生物神经系统的模拟。它的信息处理能力是由网络单元（即神经元）的输入输出特性（即激活特性）、网络的拓扑结构（即神经元的连接方式）、连接权的大小（即突触联系强度）和神经元的阀值（可视为特殊的连接权）等参数所决定的。神经网络在拓扑结构固定时，其学习过程归结为连接权的变化。在对生物神经系统进行模拟之前，我们需要对真实生物神经系统有一个大致的了解。

生物神经系统是一个有高度组织和相互作用的数量巨大的细胞组织群体。人类大脑的神经细胞大约在 10 亿个左右。神经细胞也称神经元，是组成神经系统的基本单元，数量众多的神经元按照不同的结合方式构成了复杂的神经网络。通过神经元及其连接的可塑性，使得大脑具有学习、记忆和认知等各种智能。

人工神经网络的研究出发点是以生物神经元学说为基础的。生物神经元学说认为，神经细胞即神经元是神经系统中独立的功能单元。生物神经系统包括中枢神经系统和大脑，均是由各类神经元相互连接组成。其独立性是指每一个神经元均有自己的核和自己的分界线或原生质膜。

生物神经元之间的相互连接以便让信息传递的部位被称为突触。突触按其不同信息传递机制，可以分为化学突触和电突触，其中化学突触占大多数，其神经冲动的传递借助于化学传递质的作用。生物神经元的结构如图 4 - 2 所示。

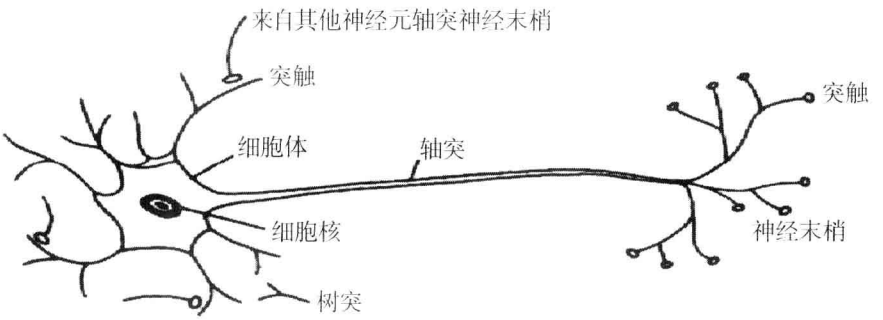


图 4 - 2　典型的生物神经元

基本神经元由细胞体和延伸部分组成。延伸部分按照功能分为两类，一类称为树突，占延伸部分的大多数，用来接收来自其他神经元传递过来的信息；另一种用来传递和输出信息，称为轴突。神经元对信息的接收和传递全部都是通过突触来进行的。一个神经元可以从别的神经元接收多至上千个突触输入。这些输入可以到达神经元的树突、胞体和轴突等不同部位，分布点各不相同，对神经元影响的程度也不同。

4.4.2　神经元功能

（1）可塑性：可塑性反映在新突触的产生和现有神经突触的调整上，可塑性使神经网络能够适应周围的环境。

（2）时空整合功能：时间整合功能表现在不同时间、同一突触上，空间整合表现在同一时间、不同突触上。

（3）兴奋与抑制：当传入神经元的冲动，经整合，使细胞膜电位升高，超过动作电位的闭值时，即兴奋状态，产生神经冲动，由轴突神经末梢传出。当传入神经元的冲动，经整合，使细胞膜电位降低，低于闽值时，即为抑制状态，不产生神经冲动。

（4）脉冲与龟位转换：沿神经纤维的脉冲为等幅、恒宽、编码（60 ~ 100mV）的离散脉冲信号，而细胞电位变化为连续信号。在突触接口处进行“数/模”转换。神经元中的轴突非常长和窄，具有电阻高、电压大的特性，因此轴突可以建模成阻容传播电路。

（5）突触的延时和不应期：突触对神经冲动的传递具有延时和不应期。在相邻的二次冲动之间需要一个时间间隔。在此期间对激励不响应，不能传递神经冲动。

（6）学习、遗忘和疲劳：突触的传递作用有学习、遗忘和疲劳的过程。

4.4.3　人工神经元的结构模型

神经元是神经网络的基本信息处理单元。它一般是一个多输入的非线性模型，其一般的结构模型如图 4 - 3 所示。

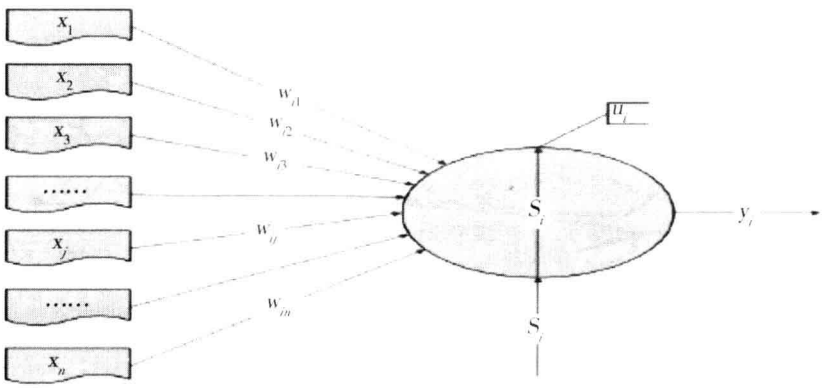


图 4-3　人工神经元

其 u_i 为神经元 i 的内部状态，θ_i 为阀值，x_j 为输入信号，w_{ij} 表示与神经元连接的权值，s_i 表示一个外部输入的控制信号。

神经元模型通常用一阶微分方程来描述（模拟生物神经网络突触膜电位随时间变化的规律），即：

$$\tau \frac{\mathrm{d}u_i}{\mathrm{d}t} = -u_i(t) + \sum w_{ij}x_j(t) - \theta_i$$

$$y_i(t) = f[u_i(t)]$$

神经元的输出由函数 f 表示。常用非线性函数来表示，例如：$f(u_i) = \dfrac{1}{1+\exp(-u_i/c)}$。其中 c 为常数。

4.4.4　人工神经网络的互联模式

根据神经元的连接方式的不同，人工神经网络的互联模式分为：

（1）前向网络

前向网络如图 4-4 所示：神经元分层排列，分别组成输入层、中间层（也称为隐含层）和输出层。每一层的神经元只接收来自前一层神经元的输入。后面的层对前面层没有反馈信号。输入模式经过各层次的顺序传输，最后在输出层上得到输出。感知器和误差反向传播算法所采用的网络均属于前向网络互联模式。

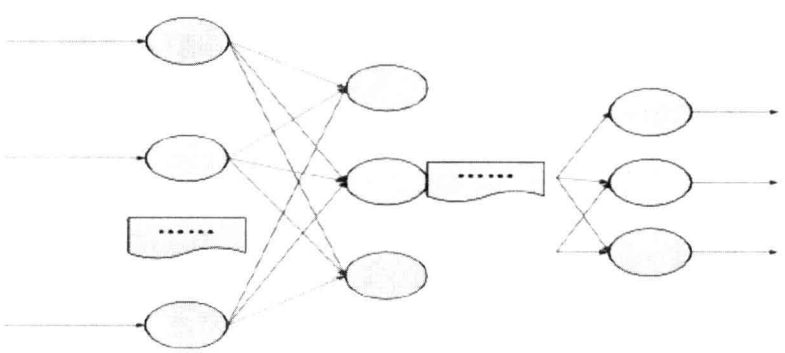


图 4-4　前向网络

（2）有向反馈的前向网络

有向反馈的前向网络如图 4 - 5 所示，从模型的输出层对输入层有信息反馈。这种网络可以用来存储某种模式序列，如神经认知机即属于此类。

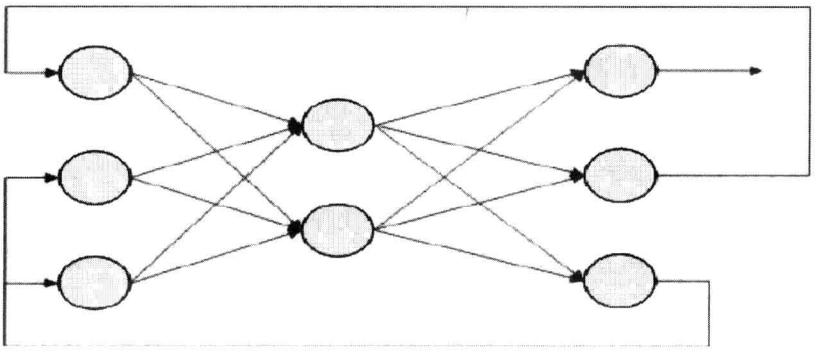


图 4 - 5　有反馈的前向网络

（3）层内有相互结合的前向网络

层内有相互结合的前向网络如图 4 - 6 所示，通过层内神经元的相互信息互联，可以实现同一层内神经元之间的横向抑制或兴奋机制。

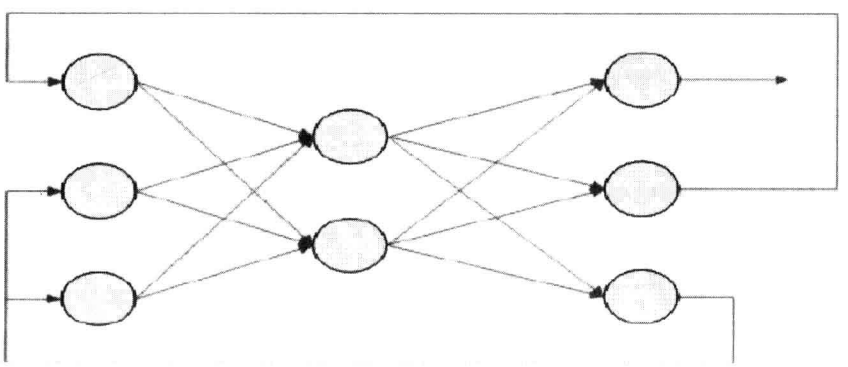


图 4 - 6　层内有相互结合的前向网络

（4）相互结合型网络

相互结合型网络如图 4 - 7 所示，这种网络在任意两个神经元之间都可能有信号连接息在神经元之间反复传递，网络处于一种不断改变状态的动态之中。

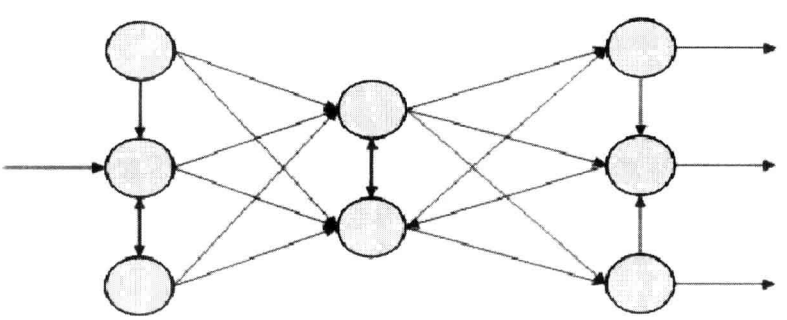


图 4 - 7　相互结合型网络

4.5 BP 神经网络

4.5.1 BP 神经网络概述

BP 人工神经网络（Back Propagation Artificial Neural Network （BPANN））是对神经系统的模拟，包括了大脑神经系统的许多特征。人脑是一个高度复杂的、非线性的和并行的处理器，它能够以比今天已有的最快的计算机还要快许多倍的速度进行计算，由于目前人类对大脑神经系统及其智能机理研究水平以及相关科学技术水平有限，因此 BP 人工神经网络对大脑神经系统进行了合理的简化和抽象，BP 人工神经网络是一个由大量结构简单的处理单元即神经元，通过广泛连接而形成的神经系统，通过大量神经元的并行计算和分布存储，BP 人工神经网络具有很强的计算能力和存储能力，因此 BP 人工神经网络是一种基于大规模并行的分布式处理系统，它的单隐层模型如图 4－8 所示共有 3 层组成。

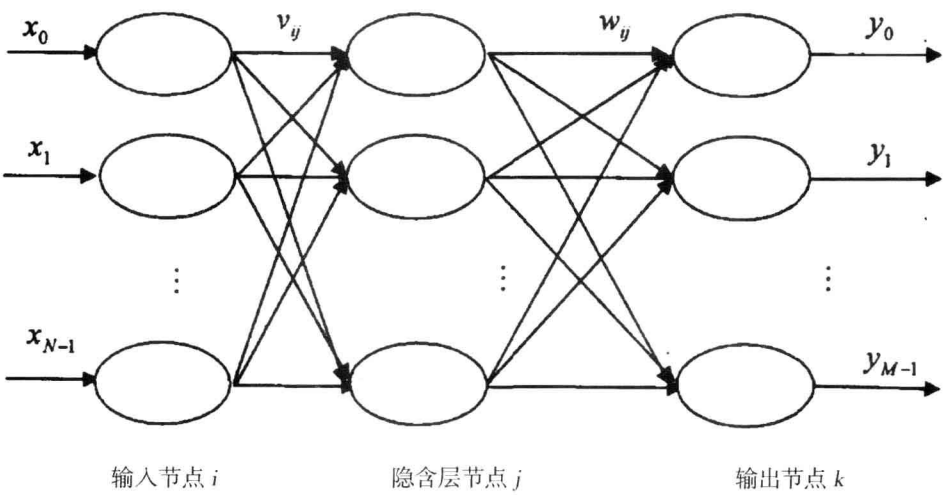


图 4－8　BP 人工神经网络

BP 人工神经网络的工作过程通常有两个阶段组成：第一阶段是学习期，在这一阶段各节点的输出保持不变，而网络的学习则是从输出层开始反向逐层逐个节点计算各连接权值的修改量，以修改各连接权值直到输入层为止；第二阶段是工作期，在这一阶段网络各节点的连接权值不变，网络的计算从输入层开始逐层逐个节点计算各个节点的输出直到输出层的各节点计算完毕。

BP 神经网络将神经网络学习输入输出的映射问题转变为非线性优化问题，使用最优化中的梯度下降算法，用迭代运算修正网络权值，实现网络输出与期望输出间的均方差最小化。算法由前向计算过程和误差反向传播过程组成。在前向计算过程中，输入信号从输入层经隐层逐层处理，传向输出层，每一层神经元的状态只影响下一神经元的状态。如果在输出层得不到期望的输出，则反向传播，将误差信号沿原来的连接通路返回，通过修改各层的权值，使得误差信号最小。权值修改量只与该权值相连的前一节点的输出和后一节点的误差有关，这样使得 BP 算法易于实现。

神经元是神经网络操作的基本信息处理单位，图 4 - 9 表示单个神经元的结构模型。它是神经网络的设计基础。下面给出神经元模型的三种基本元素：

（1）突触或连接链：每一个都由其权值或强度作为特征，神经元 k 的突触 j 上的输入信号 x_j 被乘以 k 的突触权值 w_{kj}，注意突触权值 w_{kj} 的下标的写法很重要，第一个下标指查询神经元，第二个下标指权值所在的突触的输入端，人工神经元的突触权值有一个范围可以取正值也可以取负值。

（2）加法器：用于求输入信号被神经元的相应突触加权的和，这个操作构成一个线性组合器。

（3）激活函数：用来限制神经元输出振幅，激活函数也称为压制函数，由于它将输出信号压制（限制）到允许范围之内的一定值，通常一个神经元输出的正常幅度范围可写成单位闭区间［0，1］或者另一种区间［ -1， +1］。

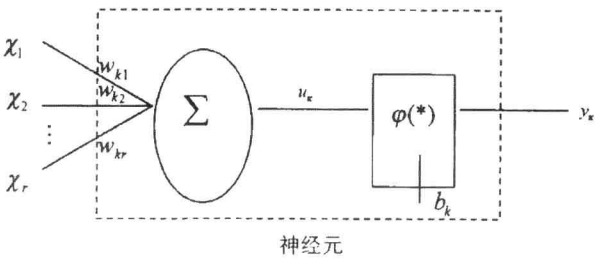


图 4 - 9　单位神经元结构模型

图 4 - 9 的神经元模型中也包括了一个外部偏置记为 b_k，偏置的作用是根据其为正或负相应地增加或降低激活函数的网络输入，其中 $p_j(j = 1,2,\cdots,r)$ 为输入信号，$w_{kj}(j = 1, 2,\cdots,r)$ 表示输入信号的权值分量。神经元输出受输入信号的影响外也受到神经元内部其他因素的影响，故在人工神经元建模中常常还加入一个额外输入信号 b，称为偏差有时也称为阀值或门限值。

4.5.2　BP 算法

（1）在样本数据输入前首先要对其进行规范化处理。这样做一方面是为了消除量纲的影响；另一方面，由于神经元的激励函数采用 Sigmoid 函数，根据其中间高增益、两端低增益的特性，当数据在远离 0 的区域里进行学习时，由于在饱和区段，学习收效速度很慢，甚至会出现麻痹现象。为此，在网络学习前，先对全部输入、输出因子做如下规范化处理：

$$x' = 0.1 + 0.8 \times \left(\frac{x - x_{\min}}{x_{\max} - x_{\min}}\right) \qquad \text{（公式 4 - 1）}$$

其中：$x_{\max}$，$x_{\min}$为每组因子最大值与最小值，x'为因子 X 规范化后的值。

这样输入输出，数据将全部在（0，1）之间，可以大大加快网络学习速度，而数据间的联系并不减少。同时，对全部连接权值和阀值赋予（ -1，1）区间的随机数。

（2）将规范化的第 k 组训练样本 $\{a_i^k\}$ 和通过输入层传递到隐含层各节点，则隐含层节点 j 的加权输入为 s_j^k：

$$s_j^k = \sum_{i=1}^{n} W_{ij} a_i^k \qquad \text{（公式 4 - 2）}$$

加权输入 s_j^k 经隐含层节点的传递函数作用后可得到隐含层节点 j 的输出 b_j^k。

$$b_j^k = f(s_j^k + \theta_j) = f\left(\sum_{i=1}^{n} W_{ij} a_i^k \theta_j\right) \quad (公式4-3)$$

式中：w_{ij} 表示输入层节点 i 至隐含层节点 j 的连接权 $i = 1, 2, \cdots, n$，$j = 1, 2, \cdots, p$；θ_j 为隐含层节点 j 的阀值；传递函数 $f(\square)$ 为 Sigmoid 函数；$k = 1, 2, \cdots, m$，m 为训练样本的组数或学习模式对数。

（3）隐含层的输出 b_j^k 作为输出层的输入传递到输出层各节点，经输出层的传递函数 $f(\square)$ 变换后得到输出层各节点的输出 c_i^k，即：

$$c_i^k = f\left(\sum_{j=1}^{p} v_{jt} + \gamma_t\right) \quad (公式4-4)$$

式中：v_{jt} 表示隐含层节点 j 至输出层节点 t 的连接权，$t = 1, 2, \cdots, q$，为输出层节点 t 的阀值；传递函数仍为 Sigmoid 函数。

（4）用期望输出 y_t^k、网络实际输出 c_t^k，计算输出层各单元的一般化误差 d_t^k。

$$d_t^k = (y_t^k - c_t^k) c_t^k (1 - c_t^k) \quad (公式4-5)$$

（5）用连接权 v_{jt}、输出层的一般化误差 d_t^k 和隐含层的输出 b_j^k 对计算隐含层各单元的一般化误差 e_j^k。

$$e_j^k = \left(\sum_{i=1}^{q} d_t^k v_{jt}\right) \cdot b_j^k \cdot (1 - b_j^k) \quad (公式4-6)$$

（6）用输出层各单元的一般化误差 d_t^k、隐含层各单元的输出 b_t^k、修正连接权 v_{jt} 和阀值 γ_t。式中：η 为学习速率。

$$v_{jt}(N+1) = v_{jt}(N) + \eta d_t^k b_t^k \quad (公式4-7)$$

$$\gamma_t(N+1) = \gamma_t(N) + \eta d_t^k \quad (公式4-8)$$

（7）用隐含层各单元的一般化误差 e_j^k、输入层各单元的输入 a_t^k、修正连接权 w_{ij} 和阀值 θ_j。

$$w_{ij}(N+1) = w_{ij}(N) + \eta e_j^k a_t^k \quad (公式4-9)$$

$$\theta_j(N+1) = \theta_j(N) + \eta e_j^k \quad (公式4-10)$$

（8）随机选取训练样本集中下一个学习模式对提供给网络，重复步骤（3）～（8），直至全部 m 个模式对训练完毕，即完成了训练样本集的一轮训练。

（9）计算网络全局误差函数 E。

$$E(W) = \sum_{k=1}^{m} E(W_k) = \frac{1}{2} \sum_{k=1}^{m} \sum_{t=1}^{q} (y_t^k - c_t^k)^2 \quad (公式4-11)$$

如果 E 小于预先设定的一个较小值或已达到设定的最大训练次数，则停止学习；否则重复步骤（3）～（9），进行样本集的一下论训练。

（10）对已经训练好的 BP 网络加载检验样本，输出预测结果。

在以上的 BP 网络学习过程中，步骤（2）、（3）作为输入信息的“正向传递过程”，步骤（4）～（7）为网络误差的“反向传播过程”，步骤（8）、（9）则是完成训练和收敛的过程，步骤（10）是对训练好的 BP 网络的应用。

BP 算法的实现流程如图 4-10 所示。

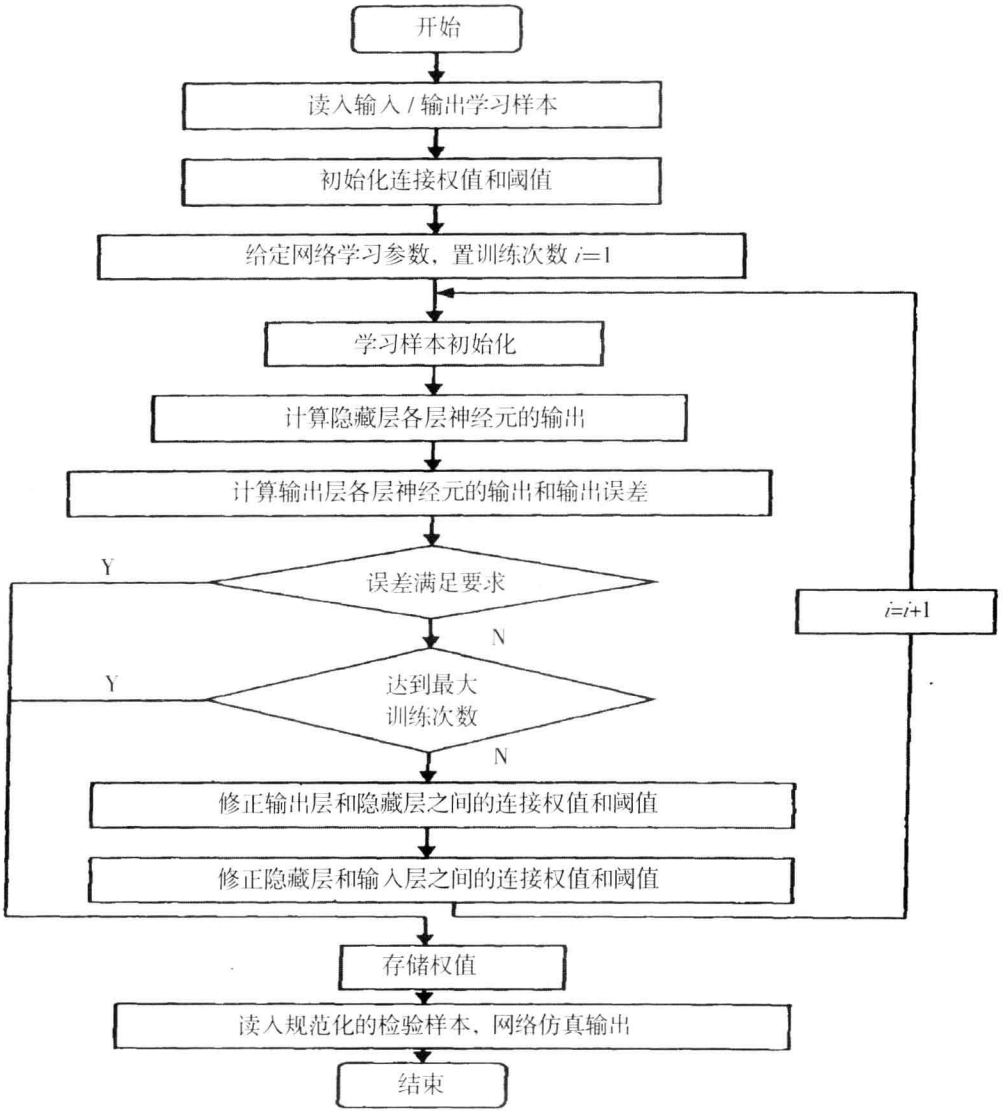


图 4 - 10　BP 算法流程图

4.5.3　BP 算法的缺陷

基于 BP 算法的神经网络从运行过程中的信息流程上看，它是前馈型网络。这种网络仅通过许多具有简单处理能力的神经元的复合作用使网络具有复杂的非线性映射能力而没有反馈，因此它不属于一个非线性动力学系统，而只是一个非线性映射。尽管如此，由于它在理论上的完整性和成功地应用于广泛的问题，所以它仍然有重要的意义，但它也存在不少问题，归纳起来有 5 点：

（1）学习算法的收敛速度很慢。需要较长的训练时间。对于一些复杂的问题，BP 算法可能需要进行几次训练才能收敛，占用较长的机时。这主要是由于学习速率太小造成的。

（2）已学习好的网络的推广（泛化）问题。即能否逼近规律和对于大量未经学习过的输入矢量也能正确地处理，并且是否具有一定的预测能力。

（3）网络隐层神经元个数的选取尚缺少统一而完整的理论指导（即没有很好的解析式来表示）。

（4）网络易陷入局部极小点。BP 网络的误差函数曲面是“凸凹不平”的，可能有许多“小坑”和一个“最低点”，也就是说，可能存在许多局部极小点和一个全局最小点。基本 BP 算法采用最速下降法，训练从误差曲面的某一初始点开始，沿曲面下降的方向滑向曲面的某一“谷底”。这个“谷底”可能是全局最小点，也可能是局部极小点，因此，网络训练容易陷入局部极小点。

（5）网络训练容易陷入 S 型函数的饱和区。如果神经元采用 S 型激励函数，当权值太大或学习率太大时，可能使网络计算陷入 S 型激励函数的饱和区。在这种情况下，S 型函数的导数很小，导致权值和偏差的修正值也很小。如果 S 型函数的导数趋向零，则权值和偏差的修正值也趋向零，从而导致网络对权值和偏差的调节作用几乎停顿下来，这种现象称为“麻痹现象”。

网络的学习过程分正向传播和反向误差传播两个部分，将需要识别的模式加到输入节点后，由网络的前向通道传播至各隐含层节点，经隐含层节点的传递函数作用后，传播到输出层的各个节点，再经输出节点的传递函数输出结果，从而完成对输入模式的识别。

4.6 BP 神经算法应用实例

将 BP 神经网络算法应用与风电场功率预测的研究。将所构建的模型应用于风电场功率预测实际算例中，检验其可行性，并分析实验结果。

4.6.1 建立数据库

神经网络应用的前提是要有足够的训练与测试数据，因此，为方便后续计算，首先对数据进行整理，建立数据库。数据来源主要有两个：一是我国北方某风电场的每十分钟一记录的平均风速、风向和实测功率值等参数值；二是数值天气预报预测的每十分钟的风速、风向和温度等参数值。

首先要对所选数据的合理性进行检验，检验标准依据国标 GB/T18710－2002，风速应为大于零的数据，风向应在 0°～360°之间，对异常与缺测数据进行适当的处理。各测量参数的合理范围及合理变化趋势如表 4－1 和表 4－2 所示。

表 4－1 主要参数的合理范围参考值

主要参数	合理范围
平均风速	0≤小时平均值＜40m/s
风向	0°≤小时平均值＜360°
气压（海平面）	94kPa＜ P ＜106kPa

表 4－2　主要参数的合理变化趋势参考值

主要参数	合理范围
1h 平均风速变化	<6m/s
1h 平均温度变化	<5°
3h 平均气压变化	<1kPa

功率数据的具体处理原则如下：

①将十分钟数据简化整理为每小时的平均值；

②缺测数据应以前后相邻时刻的数据进行插补；

③大于装机容量的实测功率值以装机容量功率值代替；

④功率小于零的实测功率值以零代替；

⑤其余不合理数据值应以前一时刻数据值替代；

⑥对各个机组每天每时刻的功率值进行汇总，得到此风电场每时刻的功率。

对以上更改替换的数据用不同颜色用于区别下面给出的具体示例。

功率随时间变化的曲线图如图 4－11 中所示，在红色圆圈所示的位置功率出现负值，依据处理原则应将其处理为零，绿色圆圈所示的位置功率值大于 1500kW，即超出单台机组的额定出力，因此将其处理为 1500kW。

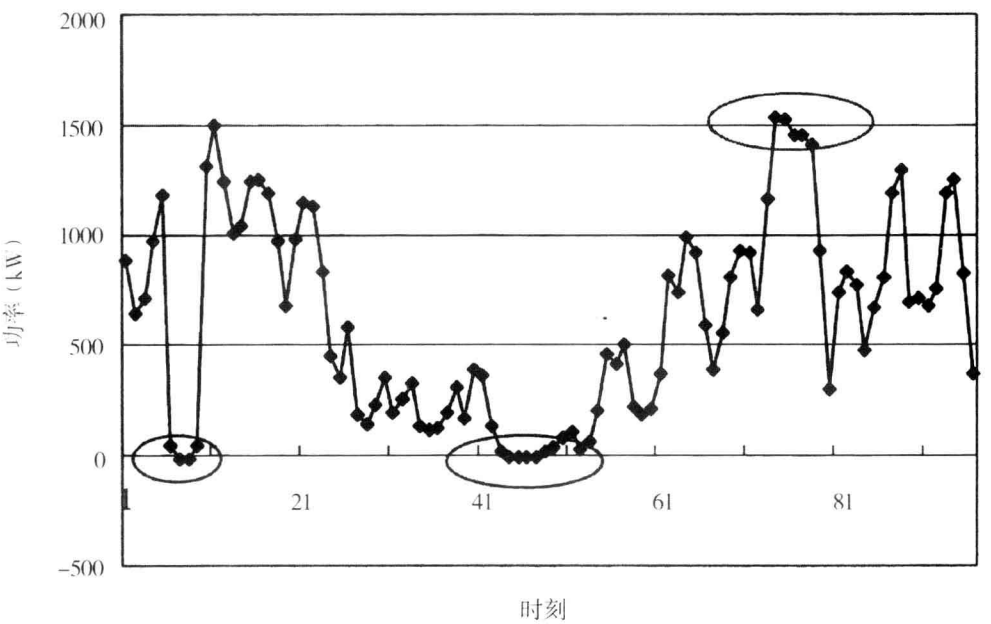


图 4－11　功率实测值示意图

表 4－3 为数据库中截取的风电场在 3 月 1 日某台机组的实测数据整理示例，其他组数据同理进行整理。表 4－4 是针对数值天气预报所建立的数据库中截取的整理示例。

表 4-3　风电场在 3 月 1 日某台机组的实测数据整理示例

时刻	风速（m/s）	风向（°）	功率（kW）
0：00	8.73	359.81	890.78
1：00	7.45	354.87	582.59
2：00	7.69	1.57	614.23
3：00	9.31	2.44	1011.35
4：00	9.86	0.45	1082.63
5：00	9.88	357.64	1108.89
6：00	8.68	2.02	744.69
7：00	9.26	1.88	932.35
8：00	8.80	5.10	836.21
9：00	10.37	359.61	1208.74
10：00	12.21	1.06	1497.58
11：00	10.54	0.14	1262.60
12：00	10.13	0.88	1166.93
13：00	10.33	0.78	1222.09
14：00	10.92	2.13	1349.75
15：00	11.01	0.28	1352.46
16：00	10.78	0.59	1297.29
17：00	9.60	358.14	1014.94
18：00	8.69	1.22	766.91
19：00	9.50	359.26	982.05
20：00	10.07	358.35	1163.82
21：00	10.55	4.45	902.82
22：00	9.42	1.60	967.46
23：00	7.90	358.94	594.81

表 4-4　数值天气预报在 3 月 2 号的部分数据整理示例

时刻	风速（m/s）	风向（°）	温度（℃）
0：00	6.34	308.74	-12.44
1：00	6.03	312.03	-12.89
2：00	6.14	315.86	-13.27
3：00	6.50	318.00	-13.59
4：00	6.39	321.74	-13.87
5：00	6.94	325.75	-14.10
6：00	6.20	329.00	-14.27

（续表）

时刻	风速（m/s）	风向（°）	温度（℃）
7：00	5.44	332.30	-14.37
8：00	4.53	333.65	-12.73
9：00	3.93	338.66	-11.63
10：00	2.97	238.50	-10.58
11：00	2.96	36.71	-9.26
12：00	3.20	66.06	-8.10
13：00	2.10	80.61	-7.30
14：00	1.20	179.86	-6.90
15：00	2.38	239.42	-6.51
16：00	3.56	254.98	-6.47
17：00	3.13	240.09	-6.69
18：00	2.55	206.88	-7.06
19：00	2.46	226.40	-7.54
20：00	3.25	249.21	-7.99
21：00	3.61	248.09	-8.02
22：00	3.03	239.32	-7.85
23：00	3.74	265.97	-7.59

4.6.2 BP 神经网络建模

BP 网络是一种多层前馈型神经网络，其神经元的传递是 S 型函数，输出量为 0～1 之间的连续量，它可以实现从输入到输出的任意非线性映射。由于权值的调整采用反向传播学习算法，因此也常称其为 BP 网络（Back Propagation Network）。

（1）输入参数的确定

风力发电机组所获得的风功率可用如下公式表示：

$$P = \frac{1}{2} C_p \rho A v^3 \qquad \text{（公式 4-12）}$$

式中：P 为输出功率，kW；C_p 为风力发电机组的功率系数；ρ 为空气密度，kg/m^3；A 为风轮扫风面积，m^3；v 为风速，m/s。

①风速与风电机组输出功率的关系

从公式 4-12 中可知，风电机组输出功率 P 与风速 v 的三次方成正比，如图 4-12 所示，风速的微小变化反应到风电机组的输出功率上将会有很大的变化，所以风速是影响输出功率的主要因素之一。

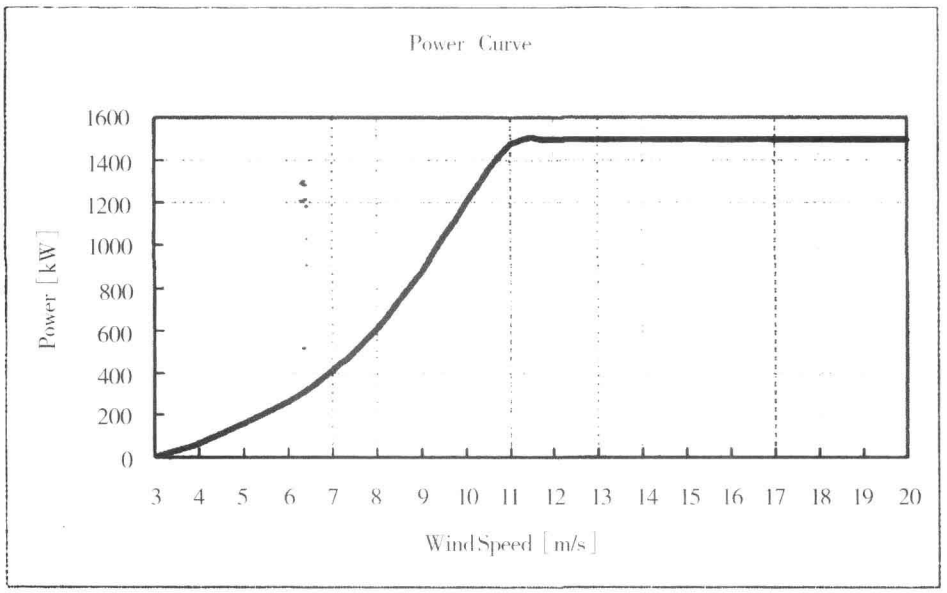


图 4 - 12　某风电机组的功率曲线（额定容量 1.5MW）

② 风向与风电机组输出功率的关系

风电场由很多风电机组组成，由于风能被风电机组的风轮吸收，风流过风轮后的风速降低，这就是尾流。由于上风向的风电机组尾流对下风向风电机组的影响，使得下风向风电机组捕获的风能减少，相应风电机组的出力也降低。如图 4 - 13 所示，相同的风速下对应机组的功率值不是像图 4 - 12 所示的一个值，而是有一簇值，这与风电机组是否对准来风方向有关，当正对来风方向时，风电机组的输出功率值比较靠近设计功率值，当有一定的角度是，风能不能被风电机组全部吸收，风电机组的输出功率小于设计值。所以风向也是影响输出功率的主要因素之一。

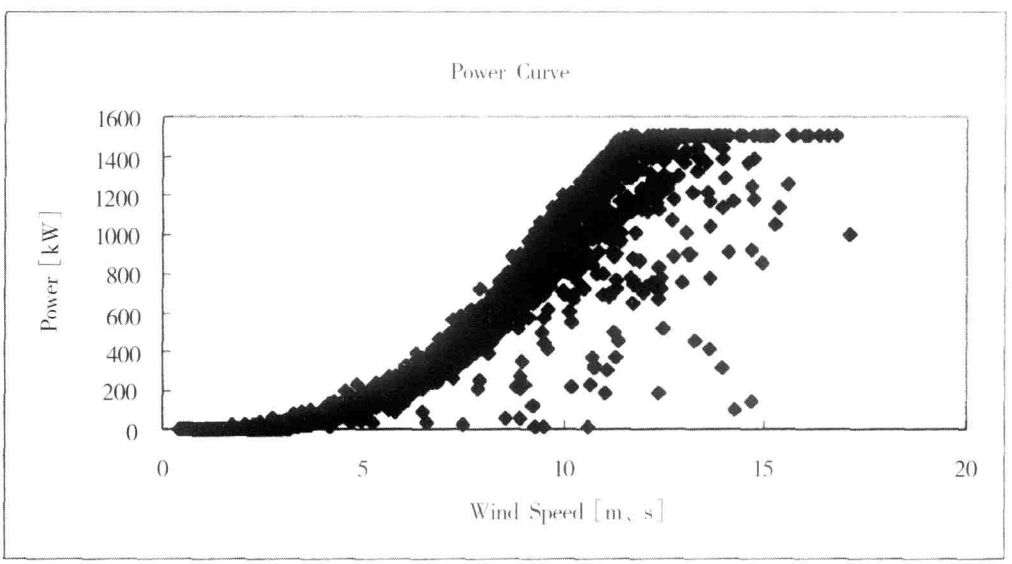


图 4 - 13　某机组实测功率散点图

③空气密度与风电机组输出功率的关系

空气密度 ρ 也是影响输出功率的重要因素之一。随着空气密度的增大，风电机组的输出功率也相应变大。而湿度、温度、压强与空气密度密切相关。因此在风电功率预测中，温度、湿度和压强也是要考虑的因素。

（2）网络结构的建立主要包括以下几个方面：

①网络隐层结构

对于三层 BP 神经网络，许多人已经证明其能够解决复杂的非线性问题，当然，有些问题如果增加层数是可能进一步降低误差、提高精度的，但同时也必然使网络复杂化。为求简化结构，本文采用单隐层的网络结构，如图 4－14 所示。

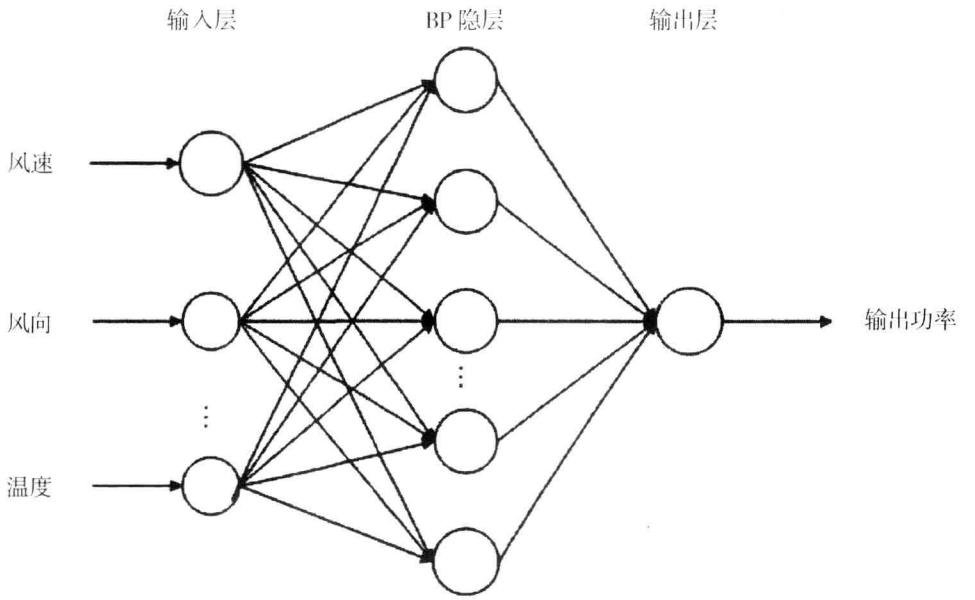


图 4－14　BP 网络结构图

②隐层神经元数的选择

在隐层数目确定的情况下，适当增加一定数目的神经元数可以提高网络训练的精度。不过也存在以下两种情况：①神经元数目过少，网络学习效果差，训练次数较多，时间较长，预测精度较低。②神经元数目过多时，也会产生其他的不利影响，精度同样不高。如何确定最适合的神经元数目是关键，针对此问题，采用将神经元数目循环叠加的方式，即从 0 开始每计算一次增加一个神经元，比较不同神经元数目下的结果，从而得到最佳神经元数。

③输出层神经元确定

假设预测提前 6 小时的功率，通常输出神经元设计为 6 个，即 6 个不同时刻的功率值。本节预测提前 24 小时的风电功率，传统的做法是应有 24 个输出神经元，作者采用设立一个输出神经元而运算 24 次的方法，大大简化了模型结构，且缩短了运算时间。

（3）传递函数

传递函数选取是否恰当会直接影响算法能否实现与预测结果的优劣，经比较研，网络隐层神经元传递函数选用 S 型正切函数，输出层神经元传递函数采用 S 型数函数。

（4）BP 算法步骤

BP 算法程序在 Matlab7.5 平台上进行调试计算：

①准备训练和测试所需用的数据，对其进行归一化处理，针对不同的数据采不同的归一化方法，将原始数据转换至［0，1］区间。

②确定隐层神经元数目。

③网络进行训练学习，以达到所设定误差值为学习目标。

④使用测试样本对风电场功率进行预测。

图 4－15 给出具体的 BP 神经网络模型预测流程图。

4.6.3 BP 神经网络实例分析

（1）隐层神经元数目的影响

BP 神经网络模型的一个重要影响因素是隐层神经元数目的选取。通常，如果神经元数目太少，BP 神经网络不能处理复杂的映射关系，网络预测误差较大。如果神经元数目过多，有可能出现“过拟合”现象，网络的预测能力也会下降，预测误差同样较大。作者首先固定训练数据与测试数据量，前 7 个月的数据用于训练，后个月的数据用于测试，然后调整神经元数目，比较不同神经元数目下的 BP 神经网络的预测误差，如表 4－5 所示。

表 4－5　不同神经元数目 BP 神经网络预测误差

隐层神经元（m）	均方根误差 RMSE（%）	平均绝对误差 MAE（%）	运行时间 TIME（s）
3－4－1	14.25	11.97	33.41
3－5－1	14.19	11.11	27.66
3－6－1	14.98	12.17	14.08
3－7－1	14.35	11.49	30.25
3－8－1	14.41	11.28	23.28
3－9－1	14.22	11.23	27.89
3－10－1	15.94	12.78	30.89

由于输入变量较少，结构简单，所以在神经元数目为 5 时，模型表现了很好的预测结果：均方根为 14.19%，平均绝对误差为 11.11%，同时时间也相对较短。当继续增加神经元数目时，网络学习能力下降，所用时间也随之增加。

（2）训练样本对预测精度影响

BP 神经网络预测的精度与训练样本的量有一定的关系，尤其对于输入参数为多变量的网络，如果缺乏足够的网络样本数据，网络预测精度可能出现较大的误差，流程图如 4－15所示。如果训练数据过多，网络运算时间过长，且出现“过拟合”现象，同样精度较低。因此，选取合适的网络训练数量是很关键的。作者针对此因素进行了实验，具体做法是将训练数据从 7 个月渐渐调整为 4 个月（如表 4－6～表 4－8 所示），测试数据不变，比较其不同预测误差值。

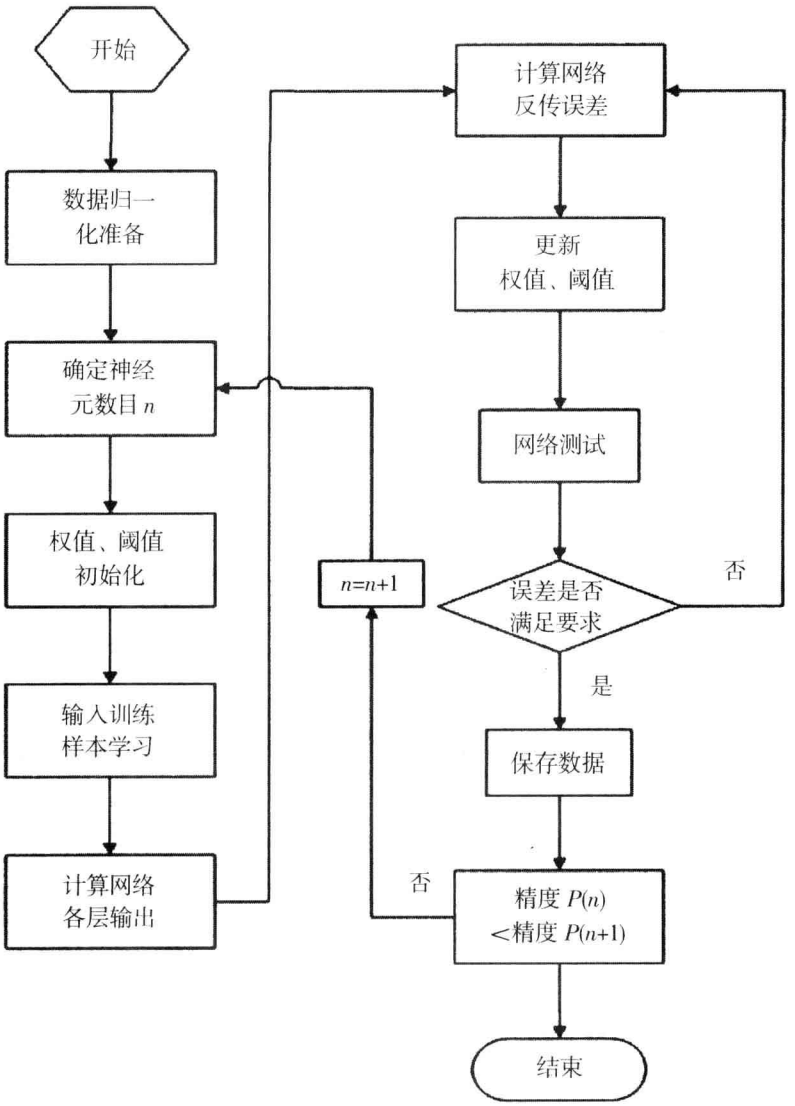


图 4-15　BP 神经网络模型预测流程图

表 4-6　6 个月训练样本 BP 神经网络预测误差

网络结构	均方根误差 RMSE（%）	平均绝对误差 MAE（%）	运行时间 TIME（s）
3-4-1	14.45	11.38	23.34
3-5-1	14.87	11.71	20.02
3-6-1	14.30	11.12	20.02
3-7-1	14.14	10.94	21.39
3-8-1	14.09	10.86	23.97
3-9-1	14.28	11.08	23.13
3-10-1	14.22	11.22	19.75

表 4-7　5 个月训练样本 BP 神经网络预测误差

网络结构	均方根误差 RMSE（%）	平均绝对误差 MAE（%）	运行时间 TIME（s）
3-4-1	14.94	11.78	17.77
3-5-1	14.85	11.48	15.53
3-6-1	14.86	11.71	15.59
3-7-1	14.49	11.25	16.16
3-8-1	14.47	11.18	11.39
3-9-1	14.04	10.85	28.86
3-10-1	14.22	11.02	19.94

表 4-8　4 个月训练样本 BP 神经网络预测误差

网络结构	均方根误差 RMSE（%）	平均绝对误差 MAE（%）	运行时间 TIME（s）
3-3-1	14.60	10.78	14.78
3-4-1	14.58	10.71	11.84
3-5-1	15.56	11.47	13.82
3-6-1	16.08	11.81	19.39
3-7-1	14.85	10.84	17.09
3-8-1	15.51	11.59	16.11
3-9-1	16.68	12.78	13.69

分析比较表 4-5～表 4-8，在输入变量相同的情况下，训练样本量不同精度较高的隐层神经元数目也不同。不同训练量的最佳预测误差汇总如图 4-16 所示，当训练数据为 5 个月时，最小均方根误差为 14.04%，平均误差为 10.85%。整体误差较小。说明此时的训练量对模型是比较适度的。其网络结构为 3-9-1，表示 3 个输入参数，9 个隐层神经元，1 个输出。实验结果充分证明了训练样本量对预测精度有较大影响，当训练样本为 5 个月（5～9 月）数据时所建立的模型具有较高的泛化能力，其依据训练数据架构的非线性空间曲面很好地表达了输入参数与输出之间的关系，当测试数据类似或者不同于训练数据网络都能够给出合理的输出。当训练样本为 7 个月（3～9 月）时，此时训练样本较多，但预测结果不是很理想，这是通常讲的“过拟合现象”，也就是说，在网络的学习过程中，学习的样本过多容易导致只记住个别的特例或者是含有噪声的样本，从而未能学到真正的规律，降低网络的泛化能力。而当训练样本为 4 个月（6～9 月）时，误差较大，即网络可以进行学习的信息量过少，因此构建的空间曲面比较粗糙，泛化能力过低，预测能力自然较差。

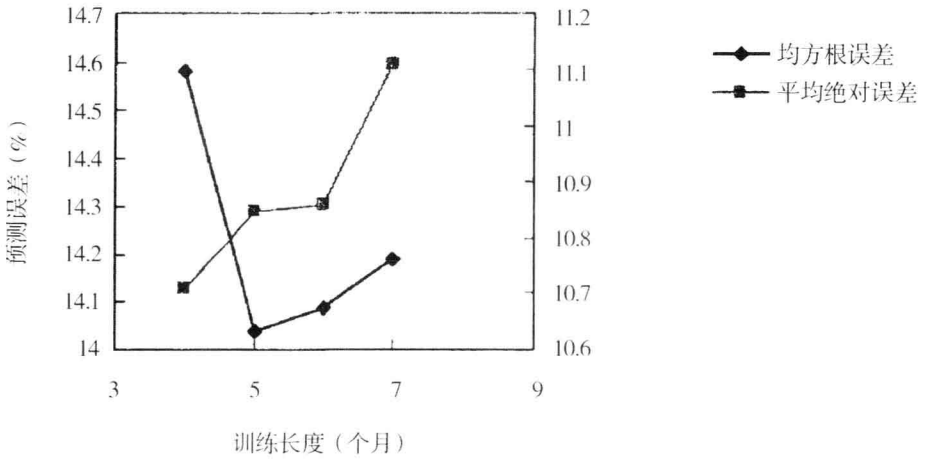


图 4－16　误差比较图

（3）输入参数的影响

在隐层神经元和训练样本确定的前提下，考虑输入参数对模型的预测能力的影响，设计操作方法为将隐层神经元数目初定为 9，训练样本为 5 个月数据，因为在此条件下，已建立的 BP 神经网络模型表现出较好的预测效果。输入变量的选取如表 4－9 所示。

表 4－9　BP 不同输入变量的预测误差

输入变量 X	均方根误差 RMSE（%）	平均绝对误差 MAE（%）	运行时间 TIME（s）
V，W	14.04	10.85	28.86
V，W，T	15.08	11.90	25.28
V，W，P	15.78	12.24	21.75
V，W，P，T	14.10	11.10	23.36
V，W，P，T，RH	14.28	11.47	20.27
V，W，P，RH	14.35	10.86	20.53
V，W，T，RH	14.16	11.38	20.08
P，T	16.76	13.75	20.43
P，T，RH	17.19	14.17	20.71

表中，V 代表风速，W 代表风向，P 代表压力，T 代表温度，RH 代表相对湿度。可以看出，输入变量中没有风速和风向时，不论均方根误差还是平均绝对误差都很大，所以，风速和风向是影响风电场功率预测的主要影响因素。输入变量中加入其他参数时，预测误差有所增加，即对于 BP 模型来说这些参数与输出之间的关系较弱。

第5章　支持向量机

5.1　概述

支持向量机（Support Vector Machine，SVM）是数据挖掘中的一项新技术，是在统计学习理论的基础上发展起来的一种新的数据挖掘方法。借助于最优化方法解决机器学习问题的新工具。它最初于20世纪90年代由Vapnik提出，近些年来在其理论研究和算法实现方面都取得了突破性进展，并开始成为克服“维数灾难”和“过学习”等传统困难的有利办法。由于其完整的理论框架和在实际应用中取得的很多好的效果，在机器学习领域受到广泛关注。

支持向量机目前主要来解决分类问题（模式识别，判别分析）和回归问题。在解决小样本非线性及高维模式分类问题中表现出许多特有的卓越性能，并能够推广应用到函数拟合等其他机器学习问题中，表现出很多优于已有方法的性能。支持向量机的出现使统计学习理论具有了有效的实现手段，引起了研究人员的重视和兴趣。由于支持向量机能在训练样本很小的情况下很好地达到分类推广的目的，且不需要预先知道分类的先验知识，因而引起了国内外学者的广泛关注。支持向量机的最大特点是改变了传统的经验风险最小化原则和结构风险最小化原则，因此具有很好的泛化能力。另外，支持向量机在处理非线性问题时，首先将非线性问题转化为高维空间中的线性问题，然后用一个核函数来代替高维空间的内积运算，从而巧妙地解决了复杂计算问题，并且有效地克服了维数灾难以局部极小问题。

5.2　统计学习理论

传统统计学研究的是样本数目趋于无穷大的渐近理论，而实际问题中的样本往往有限，因此一些理论上很优秀的方法在实际应用中不尽如人意。与传统统计学相比，统计学习理论是一种专门针对小样本的机器学习理论。在这一理论基础上发展的一种新的通用学习方法——支持向量机，最早由Cortes和Vapnik于1995年提出。其与传统学习理论最大不同在于它服从结构风险最小而非经验风险最小，是目前针对小样本估计和预测学习的最佳理论之一。它从理论上系统地研究了经验风险最小化原则成立的条件、有限样本下经验风险与期望风险的关系及如何利用这些理论找到新的学习原则和方法等问题。

统计学习理论是一种专门研究小样本情况下机器学习规律的理论。其主要内容包括以下四个方面：

（1）经验风险最小化准则下统计学习一致性的条件；

（2）在这些条件下关于统计学习方法推广性的界的结论；

（3）在这些界的基础上建立的小样本归纳推理准则；

（4）实现新的准则的实际方法（算法）。

其中最有指导性的理论结果是推广性的界，与此相关的一个核心概念是 VC 维。而支持向量机是统计学里最年轻的部分，也是最重要的部分。SVM 是 Vapnik 等根据统计学习理论中结构风险最小化原则提出的，其主要内容在 1992 ~ 1995 年间才基本完成，目前仍处于不断发展阶段。统计学习理论在 20 世纪 90 年代后越来越受重视，很大程度上是因为它发展了支持向量机。支持向量机与以往的机器学习方法相比有很多优点，解决了模型选择与过学习问题、非线性和维数灾难问题、局部极小点问题等以往困扰机器学习的很多问题。

5.2.1 VC 维

为了研究训练学习过程一致收敛的速度和推广性，统计学习理论定义了一系列有关函数集学习性能的指标，其中最重要的是 VC 维（Vapnik - Chervonenkis Dimension）。在模式识别方法中，VC 维的直观定义是：对一个指示函数集，如果存在 h 个样本能够被函数集中的函数按所有可能的 $2h$ 种形式分开，则称函数集能够把个样本打散；函数集中，VC 维就是它能打散的最大样本数目，若对任意数目的样本都有函数能将它们打散，则函数集的 VC 维是无穷大，在有界实函数中，VC 维可以通过用一定的阈值将它转化成指示函数来定义。

VC 维反映了函数集的学习能力，VC 维越大则学习机器越复杂（容量越大）。遗憾的是目前尚没有通用的关于任意函数集 VC 维的计算理论，只是对于一些特殊的函数集知道其 VC 维。比如说在 n 维实数空间中线性分类器和线性实函数的 VC 维是 $n+1$。对于一些比较复杂的学习机器（如神经网络），其 VC 维除了与函数集（神经网络结构）有关外，还受学习算法等的影响，要确定其 VC 维也就更加得困难。对于给定的学习函数集，如何（用理论或实验的方法）计算 VC 维是当前统计学习理论中有待于研究的一个问题。

5.2.2 推广性的界

统计学习理论系统地研究了对于各种类型的函数集，经验风险和实际风险之间的关系，即推广性的界。关于两分类的问题，结论是对指示函数集中的所有函数（包括使经验风险最小的函数），经验风险 $R_{emp}(w)$ 和实际风险 $R(w)$ 之间至少以 $1-\eta$ 的概率满足如下关系。

$$R(w) \leq R_{emp}(w) + \sqrt{\frac{h(\ln(2n/h)+1) - \ln(\eta/4)}{n}} \qquad \text{（公式 5-1）}$$

其中 h 是函数集的 VC 维，n 是样本数目。这一结论从理论上说明了学习机器的实际风险是由两部分组成的：一是经验风险（训练误差），另一部分称作置信范围，它和学习机器的 VC 维及训练样本数相关。可以简单地表示为：

$$R(w) \leq R_{emp}(w) + \phi(h/n) \qquad \text{（公式 5-2）}$$

它表明，在有限训练样本下，学习机器的 VC 维越高（复杂性越高），则置信范围越大，导致真实风险与经验风险之间的可能差别越大。这就是过学习现象产生的原因。机器学习过程不但要使经验风险最小，还要使 VC 维尽量小，以缩小置信范围，才能取得较小

的实际风险，从而对未来样本有较好的推广性。

需要指出的是，推广性的界是对于最坏情况的结论，在很多情况下是比较轻松的，尤其是当 VC 维较高的时候更是如此，而且这种界只有在对同一类学习函数进行比较的时候有效，指导我们从函数集中选择最优的函数，但是在不同函数集之间比较却不一定成立。因此寻找更好地反映了学习机器能力的参数和得到更紧的界是学习理论今后的研究方向之一。

5.2.3 结构风险最小化

在小样本情况下，传统机器学习方法中采用的经验风险最小化原则不合理，因为我们需要同时最小化经验风险和置信范围。在模式识别中，选定了一种分类器，就确定了学习机器的 VC 维。实际上，这种做法是先通过选择模型来选择置信范围，然后固定置信范围，通过经验风险最小化求最小风险。因为缺乏对置信范围的认识，这种选择往往是依赖先验知识和经验进行的，造成了神经网络等方法对使用者“技巧”的过分依赖。对于模式识别问题，虽然很多问题并不是线性的，但当样本数有限时，我们用线性分类器往往能得到不错的结果，其原因就是线性分类器的 VC 维比较低，有利于在样本较少的情况下得到小的置信范围。

先把函数集 $S = \{f(x,w), w \in R\}$ 分为一个函数子集序列（或叫子集结构）：

$$S_1 \subset S_2 \subset \cdots \subset S_k \subset S$$

使各个子集能够按照 h 的大小排列，也就是按照 VC 维的大小排列，即：

$$h_1 \leqslant h_2 \leqslant \cdots h_k \leqslant \cdots$$

同一个子集中，置信范围相同；在每一个子集中寻找最小经验风险，通常它随着子集复杂度的增加而减小。选择最小经验风险与置信范围之和最小的子集，就可以达到期望风险的最小，这个子集中使经验风险最小的函数就是要求的最优函数。这种方法称作结构风险最小化（Structural Risk Minimization），简称 SRM 原则，如图 5－1 所示。其中函数集 $S_1 \subset S_2 \subset S_3$，VC 维 $h_1 \leqslant h_2 \leqslant h_3$。

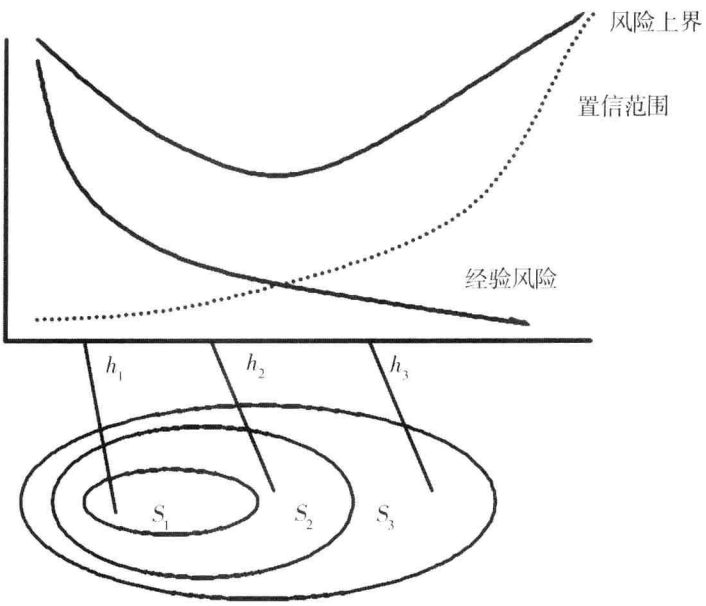


图 5－1 结构风险最小化示意图

在结构风险最小化原则下，一个分类器的设计过程包括以下两方面任务：(1) 选择一个适当的函数子集（使之对问题来说有最优的分类能力）；(2) 从这个子集中选择一个判别函数（使经验风险最小）。第一步相当于模型选择，而第二步则相当于在确定了函数形式后的参数估计。结构风险最小化原则为我们提供了一种不同于经验风险最小化的，更科学的学习机器设计原则。但是，实际上实施这一原则并不容易。如果能够找到一种子集划分方法，使得不必逐一计算就可以知道每个子集中所可能取得的最小经验风险，则上面的两步任务就可以分开进行。即先选择使置信范围最小的子集，然后在其中选择最优函数。这里的关键是如何构造函数子集结构。但是目前尚没有关于如何构造预测函数子集结构的一般性理论。

5.2.4 复杂性与推广能力

学习机器对未来输出进行正确预测的能力称为推广性。根据人们对机器学习研究的经验，最小的训练误差不一定产生最好的预测效果，其中典型的例子是神经网络的过学习问题。人们将很多注意力都集中在如何使经验风险 $R_{emp}(f)$ 更小，但很快就发现，训练误差小并不一定能得到好的预测效果。某些情况下，训练误差过小反而会导致推广能力的下降，即真实风险的增加，这就是过学习问题。

之所以出现过学习现象，一是因为样本不充分，二是学习机器设计不合理，这两个问题是互相关联的。设想一个简单的例子，假设有一组实数样本 (x,y)（不存在相同的 x），y 取值在 [0, 1] 之间，那么不论样本是依据什么模型产生的，只要用函数 $f(x,a) = \sin(ax)$ 去拟合它们（a 是待定参数），总能够找到一个使训练误差为零，但显然得到的“最优”函数 $f(x,a) = \sin(ax)$ 并不能正确代表真实的函数模型。在神经网络中，若对有限的样本来说，网络学习能力过强就会足以记住每个样本，此时经验风险很快就可以收敛到很小甚至零，但却根本无法保证它对未来样本能给出好的预测。学习机器的复杂性与推广性之间的这种矛盾同样可以在其他的学习方法中看到。由此可看出，在有限样本情况下：

①经验风险最小并不一定意味着期望风险最小；

②学习机器的复杂性不但应与所研究的系统有关，而且还要和有限数目的样本相适应。

这就是有限样本下学习机器的复杂性和推广能力之间的矛盾。有限样本情况下学习精度和推广能力之间的矛盾几乎是不可调和的，采用复杂的学习机器容易使学习误差更小，但却往往丧失了推广能力。因此，我们需要一种能够指导我们在小样本情况下建立有效的学习和推广方法的理论。统计学习理论的发展和完善，为此问题的解决提供了坚实的理论基础和有效的学习方法。

5.3 支持向量机定义

SVM 是一种准确度高的分类器，具有很好的容错和归纳能力，是一种建立在统计学理论的 VC 理论和结构风险最小原理基础上的。支持向量机技术是从线性可分情况下的最优分类面发展而来的，其最优分类线可用图 5 - 2 来说明。

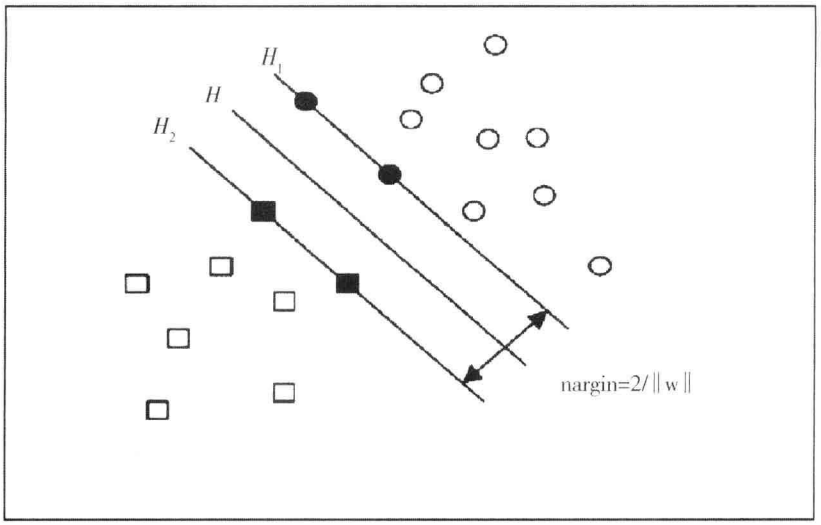


图 5 -2　线性可分情况下的最优分类线

图 5 -2 中方块和空心圆代表两类样本，H 为分类线，H_1、H_2 分别为过各类中离分类线最近的样本且平行于优分类线，且不但能保障正确地将两类数据分开，而且还要使分类间隔最大，这里把它们之间的距离叫做分类间隔（Margin）。也就是说在 H_1 右侧的被分为正类，H_2 左侧的被分为负类，位于两类中间的样本则拒绝分类。其数学表达方式如下：

设有两类线性可分样本集为 $(x_1,y_1),i=1,2,\cdots,n,x\in R^d,y\in\{+1,-1\}$ 是类别标号。d 维空间中线性判别函数的一般形式为：$g(x)=\omega\cdot x+b$，分类面方程为：

$$\omega\cdot x+b=0 \qquad \text{（公式 5 -3）}$$

将判别函数 $\omega\cdot x+b=0$ 进行归一化，使两类所有样本都满足 $|g(x)|\geqslant 1$，当 $|g(x)|=1$ 时，分类间隔就等于 $\frac{2}{\|\omega\|}$，同时也等价于使 $\|\omega\|$（或 $\|\omega\|^2$）最小；如果要求所有样本正确分类，就是要求其满足：

$$y_i[\omega\cdot x_i+b]-1\geqslant 0,(i=1,2,\cdots,n) \qquad \text{（公式 5 -4）}$$

因此，满足上述条件且使 $\|\omega\|^2$ 最小的分类面就是最有分类面。在图 5 -2 中，蓝色和红色的点就是这两类样本中离分类面最近的点，且平行于最有分类面，我们把这样的样本叫做支持向量（Support Vectors）。

根据上面的讨论，最有分类面问题可以表示成如下的约束优化问题，即在公式 5 -4 的约束下，求函数：

$$\phi(\omega)=\frac{1}{2}\|\omega\|^2=\frac{1}{2}(\omega\cdot\omega) \qquad \text{（公式 5 -5）}$$

的最小值。若作为二次规划问题来看，可定义以下的拉格朗日函数：

$$L(\omega,b,a)=\frac{1}{2}(\omega\cdot\omega)-\sum a_i\{y_i[(\omega\cdot x_i)+b]-1\} \qquad \text{（公式 5 -6）}$$

其中：$a_i>0$ 为 Lagrange 系数。对于线性不可分的样本，希望使误分类的点数最小，为此在公式 5 -4 引入松弛变量 $\xi_i\geqslant 0$，即：

$$y_i[(\omega\cdot x_i)+b]-1+\xi_i\geqslant 0,(i=1,2,\cdots,n) \qquad \text{（公式 5 -7）}$$

在公式 5 -7 中，对于给定的常数 C，求出使：

$$\phi(\omega,\xi) = \frac{1}{2}(\omega \cdot \omega) + C\{\sum_{i=1}^{n}\xi_i\} \quad \text{（公式 5-8）}$$

如果用内积 $K(x,x')$ 代替最优分类面中的点积，就相当于把原特征空间变换到了某一新的特征空间，此时油画函数变为：

$$\omega(a) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i,j} a_i a_j y_i y_j (x_i \cdot x_j) \quad \text{（公式 5-9）}$$

相应的判别函数也应变为：

$$f(x) = \mathrm{sgn}\{\sum a_i^* y_i k(x_i \cdot x) + b^*\} \quad \text{（公式 5-10）}$$

支持向量机的基本思想可以概括为：首先通过非线性变换将输入空间变换到一个高维空间，然后在这个新空间中求取最优线性分类面，而这种非线性变换是通过定义适当的内积函数实现的。

5.4 核函数

将数据映射到特征空间以便更有效地处理原空间中的任务，这一技术在模式识别领域又称作特征选择。特征选择是模式识别的一个关键问题，特征选择的好坏往往对整个系统产生决定性的作用。但如何获得针对一个特定问题的最合适的特征，目前仍然是模式识别领域中的一个难点。一般来说，由特征选择得到的特征空间的维数小于输入空间的维数，这就是通常所说的降维。因为传统的方法在处理高维问题时容易导致维数灾难，即随着特征空间维数的升高，计算复杂性呈指数增长。一个复杂的模式分类问题在高维空间会比在低维空间更容易线性可分。如果问题可以仅仅由特征空间中的内积来表示，则高维空间中特征向量的内积可以通过核函数用低维空间中的输入向量直接计算得到，从而使得计算量不会呈指数增长，避免了一般扩维方法中的维数灾难，这就是核方法的实质。使用核函数不需要事先知道映射的具体形式。事实上，给定一个核函数，总能构建一个与这个核相联系的特征空间，使得该空间中的内积可以用核函数来表示。

核函数方法就是用非线性变换可以利用核来把数据映射到特征空间，并且在这个空间训练线性机器，能够利用的训练样本的唯一信息是它们在特征空间的 Gram 矩阵。这个矩阵也称为核矩阵，用 K 表示。核函数为支持向量机（SVM）提供了一个重要的构成模块，常用的核函数有多项式函数、径向基函数和 Sigmoid 函数等，在选用不同的核函数时可以构造出不同的支持向量机，下面将介绍几个常用的核函数。

5.4.1 多项式核函数

选用下列核函数：

$$K(x,x_i) = [(x,x_i) + 1]^2 \quad \text{（公式 5-11）}$$

构造的支持向量机的判别函数为：

$$f(x) = \mathrm{sgn}\{\sum_{i=1}^{n} a_i y_i [(x_i \cdot x) + 1]^2 - b^*\} \quad \text{（公式 5-12）}$$

其中，s 为支持矢量的个数。

5.4.2 径向基核函数

选用下列核函数：

$$K(x,x_i) = \exp\{-|x-x_i|^2/\sigma^2\} \quad (公式5-13)$$

构造的支持向量机的判别函数为：

$$f(x) = \mathrm{sgn}\{\sum_{i=1}^{n} a_i \exp\{-|x-x_i|^2/\sigma^2\} - b^*\} \quad (公式5-14)$$

其中，s 个支持矢量 x_i 可确定径向基函数的中心位置，s 是中心的书目。公式 5-13 是目前被使用最多和最普遍的核函数。径向基函数所对应的特征空间可以是无穷维数，因此理论上讲，一切有限的数据样本在该特征空间中肯定是线性可分的。

5.4.3 Sigmoid 函数

选用下列核函数：

$$K(x,x_i) = \tanh(v(x \cdot x_i) + a) \quad (公式5-15)$$

其中，S 型函数采用双曲正切函数 tanh。其构造的支持向量机的判别函数为

$$f(x) = \mathrm{sgn}\{\sum_{i=1}^{n} a_i \tanh[v(x \cdot x_i) + a] + b^*\} \quad (公式5-16)$$

公式 5-16 就是通常所用的三层神经网络的判别函数，其隐节点数目、隐节点对输入节点的权重矢量 v，x_i 和输出节点对隐节点的权重 a_i 都是在模型训练过程中自动确定的。值得一提的是，Sigmoid 函数所使用的计算方法避免了人工神经网络自身所带来的局部极小点问题。

5.5 支持向量机的训练算法

两类分类的依据是数据空间中对象的距离准则，即同类对象间的距离尽可能小，不同类对象间的距离尽可能大。因为距离运算主要涉及点积运算，所以对于在低维空间中线性不可分的问题，可以通过非线性变换将低维空间变换到一个高维空间，使得在高维空间中变得线性可分，这种非线性变换正是通过核函数来实现的。经过核变换的分类函数依靠支持向量形成分类，因此，称为支持向量机。可见，核函数是支持向量机算法中重要的核心，只有通过正确的核函数才能将问题的求解由低维空间求解转化到高维空间求解。支持向量机中不同的核函数可以形成不同的算法。

支持向量机的典型训练算法主要包括选块算法、分解算法、序列最小最优化算法(SMO)。本节在分析标准支持向量机数学模型的基础上，分析了支持向量机主要的训练算法及相关子算法，包括一维搜索、最速下降法和 SMO 算法。目前，这些算法都已经有较为成熟的应用，较为常见的支持向量机仿真平台有 LibSVM 和基于 Matlab 的 SVM 仿真工具箱。

5.5.1 支持向量机模型

从前文中知道，标准支持向量机有严密的数学理论基础。从数学角度分析，支持向量

机是一个求条件极值问题，可以归结为一个二次规划优化问题的求解算法，二次规划是一种常见的优化问题，有一套比较成熟的理论。标准支持向量机就是先通过用内积函数定义的非线性变换将输入空间变换到一个高维空间，然后在这个高维空间中求取最优线性分类面的分类方法。对于原空间中线性不可分的数据，首先经过一个非线性映射 $\phi: R^d \rightarrow H, x \rightarrow \phi(x)$，映射到一个高位的特征空间 H（核空间）中，这样在核空间 H 中就可以通过核函数 k 构造最优超平面，实现数据在高维空间中的分类。因此，核函数的选择与设计至关重要，常用核函数如前文所述。通过设计核函数 $k(X_i, X_j) \leqslant \Phi(X_i), (X_j) >$ 得到支持向量机模型，如公式 5 - 17 所示：

$$\max H(a) = \sum_{i=1}^{N} a_I \sum_{j=1}^{N} y_i y_j a_i a_j k(X_i \cdot X_j) \qquad \text{（公式 5 - 17）}$$

$$\text{Subject to } \sum_{i=1}^{N} y_i a_i = 0, a_i \geqslant 0, i = 1, \cdots, N \qquad \text{（公式 5 - 18）}$$

对应的判别函数模型为公式 5 - 18 所示：

$$f(X) = \sum_{i=1}^{N} y_i a_i^* k(X_i \cdot X) + b^* \qquad \text{（公式 5 - 19）}$$

该函数的复杂度仅仅和支持向量的个数有关，而与特征空间的维数无关。“传统的统计方法只有在样本数量趋于无穷大时，才有理论的保证，而在实际应用中，样本数目通常都是有限的，甚至是小样本，基于大数定理的统计方法对此难以取得理想的效果。支持向量机基于 Mercer 核定理，寻找高维特征空间中的最优线性超平面，依靠能够形成支持向量的少数样本就可以实现分类。简单来说，支持向量机的优点就是升维、线性化和小样本。也就是说，支持向量机利用输入空间中的核函数取代了高维特征空间中的内积运算，通过判别函数使得大量的工作将在输入空间而不是在高维特征空间 H 中完成，其原理如图5 - 3 所示。

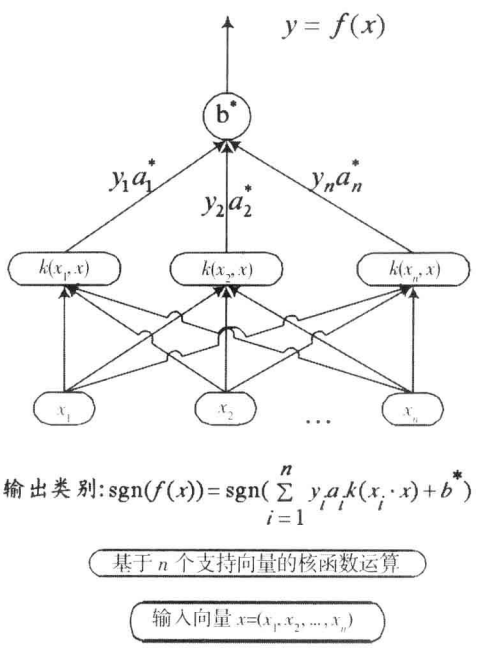


图 5 - 3　支持向量机原理图

因此，训练超平面分类方程，核函数的设计是关键，求解二次规划问题是核心。根据数学原理的分析，可总结出支持向量机算法的步骤，包括训练分类方程的步骤和分类数据的步骤两部分。

第一，支持向量机训练的步骤。

（1）输入两类训练样本向量 (X_i, X_j)（$i=1, 2, \cdots, N$，$X \in R^n$，$y \in \{-1, 1\}$），类号分别为 ω_1, ω_2。如果 $X_i \in \omega_1$，则 $y_i = -1$；如果 $X_i \in \omega_2$，则 $y_i = 1$。

（2）选择适当的核函数类型。

（3）利用二次规划方法求解公式（5-17）的最优解，得到支持向量机及最优 Lagrange 乘子 a^*。

（4）利用样本库中的一个支持向量机 X，带入公式 5-18，左值 $f(X)$ 为其类别值（-1 或 1），可得到偏差值 b^*。

第二，支持向量机分类的步骤。

（1）输入待测样本 X。

（2）利用训练好的 Lagrange 乘子 a^*、偏差值 b^* 和核函数，根据公式 5-18 求解判别函数 $f(X)$。

（3）根据 sgn $(f(X))$ 的值，输出类别。如果 sgn $(f(X))$ 为 -1，则该样本属于类 ω_1；如果 sgn $(f(X))$ 为 1 则该样品输入类 ω_2。

5.5.2 支持向量机算法

支持向量机把分类问题归结为带有线性约束的凸二次规划问题，基本算法可分为三类：第一类是把支持向量机中的最优化问题转化为无约束问题，然后求解无约束优化问题，求解此类问题的基本算法通常包括最速下降法、使用一维搜索的最速下降算法、Newton、修正 Newton 算法和共轭梯度法等；第二类是内点最优化算法，包括原仿射尺度法和原对偶算法等；第三类是求解大型问题的特殊算法，这类算法的基本思想就是将大规模的原始问题分解为多个同样性质的子问题，按照某种迭代策略，反复求解子问题，构造出原始问题的近似解，并使该近似解逐渐收敛到原始问题的最优解。选取不同的子问题或是选取不同的迭代策略会产生不同的算法，典型的有选块算法（Chunking）、分解算法和序列最优化方法（Sequential Minimal Optimization - SMO）。目前支持向量机研究和应用中，训练算法普遍采用分解算法，常见分解算法为 SVMlight、序贯最小优化（SMO）算法和 Chunking 算法。这些算法实现的核心都离不开求解最优化问题，都用到了标准二次规划问题求解对应的优化算法，只是最优化问题的具体内容不同而已。其中 SMO 算法需要解析求解两个变量的最优化问题，思路较为直观。最优化问题有一套成熟的求解算法，涉及求解支持向量机的优化算法主要有最速下降法、一维搜索算法等。

（1）最速下降法

最速下降法是基本的无约束优化问题的求解算法，将其思想推广就可以在变形后得到其他算法，具体分析如下。

设无约束问题为

$$\min f(x), x \in R^n \qquad \text{（公式 5-20）}$$

对于 n 个变量的函数 $f(x)$，它在某处点 x_K 的负梯度方向为 $d_k = -\nabla f(x_k)$。此方向是 x_k 附近使函数值下降最快的方向，因此应沿此方向前进。由 x_k 出发沿方向 d_k 前进，使所

经过的点都可以被表示为 $x_k + \alpha d_k$，其中 α 是标量。当 $\alpha = 0$ 时，$x_k + \alpha d_k$ 对应于出发点，当 α 变动时，点 $x_k + \alpha d_k$ 的轨迹是一条从 x_k 出发沿 d_k 方向的射线。为在这条直线上求出 $f(x)$ 取值更小些的点 x_{k+1}，只需要考虑单变量函数 $\varphi(\alpha) = f(x_k + \alpha d_k)$，找出一个 α_k 使 $\varphi(\alpha_k)$ 较 $\varphi(0)$ 有一定程度的下降，然后代入 $x_{k+1} = x_k + \alpha d_k$。上述对单变量函数 $\varphi(\alpha)$ 寻找 α_k 从而得到 x_{k+1} 的过程称为从 x_k 出发沿 d_k 方向的一维搜索，其中 d_k 为搜索方向。最速下降法算法过程如下：

①取初始点 x_1，令 $k = 1$；

②计算梯度 $\nabla f_k = \nabla(x_k)f$；

③若 $\nabla f(x_k) = 0$，则停止计算，得到无约束问题公式 5 - 20 的近似解 x_k，否则令 $d_k = -\nabla f(x_k)$；

④从 x_k 出发沿 d_k 方向进行一维搜索得到点 x_{k+1}；

⑤令 $k = k + 1$，转②。

同时，在求解无约束问题时，一般的下降算法都需要用到一维搜索。一维搜索就是从点 x_k 出发沿搜索方向 d_k 找出一点 x_{k+1}，使函数 $f(x_{k+1})$ 较 $f(x_k)$ 有一定程度的下降。对于单变量函数 $\varphi(\alpha) = f(x_k + \alpha d_k)$，寻找变量 α 的一个值 μ 作为步长 α_k，取 $x_{k+1} = x_k + \alpha_k d_k = x_k + \mu d_k$，就能满足要求。上述对单变量函数 $\varphi(\alpha)$ 寻找 α_k 从而得到 x_{k+1} 的过程称为从 x_k 出发沿方向 d_k 的一维搜索，其中 d_k 称为搜索方向。

（2）一维搜索算法

设已经选定搜索方向 d_k，并且已经得到单变量函数 $\varphi(\alpha) = f(x_k + \alpha d_k)$，则有一维搜索算法如下：

①取初始步长 μ_1（可取 $\mu_1 = 1$），并取精度要求 $\beta \in (0, \frac{1}{2})$（可取 $\beta = 0.1$）和步长缩放因子（$\rho \in (0,1)$ 可取 $\rho = 0.1$），置 $k = 1$；

②计算 $\varphi(\mu_k)$ 和 $\theta_k = \varphi(0) + \beta\varphi(0)u_k$；

③若 $\varphi(\mu_k) \leqslant \theta_k$（算法终止条件），则置 $\mu = \mu_k$，停止计算，得到需要计算的 μ；否则置 $\mu_{k+1} = \rho\mu_k$，$k = k + 1$，转②。

（3）SMO 算法

SMO 算法的主要思想就是每次只优化两个 α_i，可作为单变量的二次函数优化，有解析解。

第一，选两个样本点 (x_i, x_j)（$i = 1$，2 且 $j = 1$，2）对应的两个 α_i，记原始值为 α_1^{old}，α_2^{old}；

第二，对选择的这两个变量进行二次优化 $\max \sum_{\alpha_1,\alpha_2}^{2} \alpha_1 - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j)$，

$$st. \quad 0 \leqslant \alpha_i \leqslant C,\ i = 1,\ 2$$

则：

$$\alpha_1 y_1 + \alpha_2 y_2 = \alpha_1^{\text{old}} y_1 + \alpha_2^{\text{old}} y_2 \qquad \text{（公式 5 - 21）}$$

第三，求解析解。

①求解：$\alpha_2^{\text{new,unc}}, \alpha_2^{\text{new,unc}} = \alpha_2^{\text{old}} + \frac{E_1 + E_2}{k}$

其中，$k = K(x_1, x_1) + K(x_2, x_2) - 2K(x_1, x_2)$；

$g(x) = \sum_{j=1}^{1} \alpha_i y_i K(x_j, x_i) + b, E_i = g(x_i) - y_i$。

②求解 α_2^{new}，$\alpha_2^{new}=\begin{cases}V & \text{if} \quad \alpha_2^{new,unc}>V\\ \alpha_2^{new,unc}, & \text{if} \quad U\leqslant\alpha_2^{new,unc}\leqslant V\\ U & \text{if} \quad \alpha_2^{new,unc}<V\end{cases}$

其中，

当 $y_1\neq y_2$ 时，$U=\max(0,\alpha_2^{old}-\alpha_1^{old}),V=\min(C,C-\alpha_1^{old}+\alpha_2^{old})$；

当 $y_1=y_2$ 时，$U=\max(0,\alpha_2^{old}+\alpha_1^{old}-C),V=\min(C,\alpha_1^{old}+\alpha_2^{old})$。

③求解 $\alpha_1^{new},\alpha_1^{new}=\alpha_1^{old}+y_1y_2(\alpha_2^{old}-\alpha_2^{new})$

上面两个样本点 $(x_i,y_j)(i=1,2$ 且 $j=1,2)$ 的选取必须满足，在算法外层循环中，找到违反 KTT 条件的点作为第一个样本点 (x_1,y_1)，第二个样本点 (x_2,y_2) 要与第一个样本点配对，其中的一个方法是选择使得 $|E_1-E_2|$ 最大化的 (x_2,y_2)，在算法中作为内层循环。

综上所述，可得到 SMO 算法流程，如下所示的序贯最优化算法。

序贯最优化算法：

①给定精度 $\varepsilon,\alpha^0=0$，令 $k=0$；

②选优化变量 α_1^k,α_2^k，解析求解关于这两个变量的最优化问题，得最优解 $\alpha_1^{k+1},\alpha_2^{k+2}$，据此更新 α，得到 α^{k+1}；

③若在精度 ε 内满足停机条件准则

$$0\leqslant\alpha_j\leqslant C,i=1,\cdots l,\sum_{i=i}^{l}y_i\alpha_i=0,$$

$$y_i\left(\sum_{i=1}^{l}\alpha_iy_iK(x_i,x_j)+b\right)\begin{cases}\geqslant 1 & \{x_j\ \alpha_j=0\}\\ =1 & \{x_j\ 0<\alpha_j<C\};\\ \leqslant 1 & \{x_j\ \alpha_j=C\}\end{cases}$$

则转第④步；否则，令 $k=k+1$，则转第②步；

取近似解 $\alpha^*=\alpha^{k+1}$。

5.6 支持向量机应用实例

金融时间序列特指各种不同金融产品的时间序列，比如汇率、基金、股票价格等。金融时间序列是一种特殊的时间序列，它与金融市场和人类的各种经济活动紧密相关。与其他的时间序列相比，金融时间序列具有以下特征：信噪比低、信噪难以有效分离；具有显著的非平稳特性；丰富的潜在周期特性；样本数据少且维数高。时间序列预测是一种定量分析方法，它是在时间序列变量分析的基础上，运用一定的数学方法建立预测模型，使时间趋势向外延伸，从而预测未来市场的发展变化趋势，确定变量预测值。总的来看，目前国内外关于证券价格波动研究的方法主要有两大类：一是传统的统计学方法，另一类就是混沌等非线性系统的理论和方法。后者是前者的发展和补充，由于它能指出一些传统方法无法解释的“巨涨巨落”现象的本质和有序与无序的关系。

金融市场数据的非线性、非平稳性、高信噪比和有限样本，使传统的模型受到了各种挑战，SVM 的应用逐渐成为各国研究者的研究重点，且在金融时间序列的分析和预测方面做出大量有价值的研究。本节要讨论将 SVM 在上海证券综合指数（以下简称上证综指）预测方面的应用，从而来对证券价格进行预测和风险监测。

（1）实验数据选取

以上海证券交易所综合指数日线为例，来进行基于 SVM 的时间序列预测训练样本（取 2005 年 1 月至 2008 年 8 月间共 700 个样本），用于建立上证综指的 SVM 预测模型。测试样本为 2008 年 8 月 22 日至 2009 年 11 月 23 之间共 300 天的上证综指，用于评价所建立的时间序列模型能否较好地预测指数波动。

（2）数据预处理

预测过程要经过选取数据、设定模型参数、数据预处理、训练并求解支持向量、代入测试集数据计算模拟值、数据后处理、计算评价指标并对结果进行分析等过程。为了便于建立 SVM 模型，首先需要对训练样本时间序列数据进行特征提取。对于训练样本，本文选取输出为上证综指下一日的收盘值，输入为当日收盘值、5 日指数移动平均、10 日指数移动平均、20 日指数移动平均及成交量。

此外，对于一个 m 维的样本数据里，不同分量的数据处在不同的数量级上，数量级大的分量对模型的影响就较大，而数量级小的分量对模型的影响就小，小数“淹没”在了大数里而失去了意义，所选择的原始数据需要进行一定的处理，将数据标准化。本文对所有数据均采用统计方法归一化，作为 SVM 的训练样本，即：

$$x'_i = \frac{x_i - \bar{x}}{\sqrt{\mathrm{var}(x)}}, i = 1,2,\cdots,l \qquad \text{（公式 5-22）}$$

其中，$\bar{x}$，$\sqrt{\mathrm{var}(x)}$ 分别为变量 x 的平均值核标准差。

（3）支持向量机模型选择

支持向量机选择高斯径向核函数公式 5-13 作为核函数。模型参数中，$\varepsilon = 0.01$；

正则化系数 $C = 100$，采用本章第 5 节介绍的 SVM 寻优算法来建立 SVM 回归模型。

（4）预测结果分析

如图 5-4 所示，为 SVM 预测结果与实际值的比较。实线为股市实际运行结果，而虚线为本文所建立的 SVM 支持向量机预测的下一日指数结果。从图中可以看出，所建立的 SVM 预测模型能很好地预测上证综指，能够比较好地预测金融时间序列的波动。

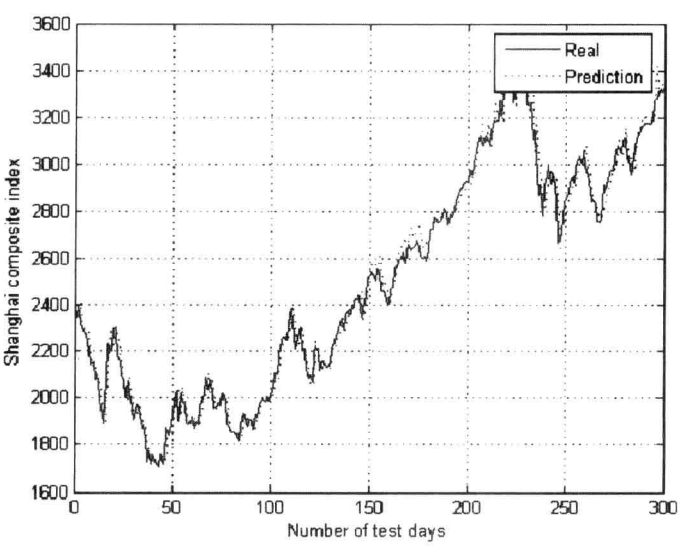


图 5-4　上证综指实际值与预测值的比较

第6章　关联规则

在数据挖掘领域，采用关联规则在大型数据库中进行数据挖掘是一个重要的研究内容。关联规则是美国 IBM Almaden Research Center 的 RabeshAgrawal 等人于 1993 年首先提出的 KDD 研究中的一个重要课题。关联规则挖掘的一般对象是事务数据库，这种数据库主要应用在零售业，比如超级市场的销售管理。关联规则挖掘就是发现事务数据库中不同商品之间是否存在某种关联关系。通过这些规则找出顾客购买行为模式，如购买了某一商品对购买其他商品的影响。发现这样的规则可以应用于商品货架设计、存货安排以及根据购买模式对用户进行分类。

譬如在商场中，许多决策只停留在管理人员的经验判断层次上，缺乏深层次的分析，也往往跟不上客观环境的变化，所以就需要对客观实时数据进行分析，找到数据的内在联系，从而指导商家制订销售计划，方便顾客购物等。关联规则的一个典型例子是购物篮分析。该过程通过发现顾客放入其购物篮中不同商品之间的联系，分析顾客的购买习惯。通过了解哪些商品频繁地被顾客同时购买，这种关联的发现可以帮助超市经营者制定营销策略。例如，在同一次去商场超级市场，如果顾客购买牛奶，同时也购买面包的可能性有多大？使用关联规则通过对过去销售数据的分析，可以帮助超市经营者有选择地经销和安排货架，这种信息可以引导销售。如将牛奶和面包尽可能放近一些，这样就方便了顾客的购买，同时也可以进一步刺激一次去超市同时购买这些商品。

另一个比较典型的案例是“尿布与啤酒”的故事。在美国的一个大型超市中，统计人员发现了一个很有趣的现象，在各种产品组合销售中，同时购买啤酒和尿布的组合竟然占据了大多数。这种奇怪的组合象引起了销售人员的注意，于是他们有意把啤酒和尿布摆在一起出售，结果销量竟然成倍增加了。这不是一个玩笑，而是真实地发生在了美国的大型连锁超市“沃尔玛超市”中。分析人员就此展开了一系列的调查活动，终于找到了问题的根。原来，在美国，一些年轻的父亲下班后经常要到超市去给孩子买尿布，但是他们当中30% ~40%的人都有喝酒的嗜好，因此在购买婴儿尿布的同时也为自己买了啤酒，这才导致啤酒和尿布的搭配销售增加了各自的销量。沃尔玛拥有世上最大的数据仓库系统，为了能够准确了解顾客在其门店的购买习惯，沃尔玛对其顾客的购物行为进行购物篮分析，想知道顾客经常一起购买的商品有哪些。沃尔玛数据仓库里集中了其各门店的详细原始交易数据。数据关联是数据库中存在的一类重要的可被发现的知识。若两个或多个变量的取值之间存在某种规律性，就称为关联。关联可分为简单关联、时序关联、因果关联。关联分析的目的是找出数据库中隐藏的关联网。有时并不知道数据库中数据的关联函数，即使知道也是不确定的，因此关联分析生成的规则带有可信度。

6.1 关联规则相关定义及基本概念

定义 6.1 关联规则

给定一组项集 $I = \{i_1, i_2, \cdots, i_k\}$ 和一个事物数据库 $D = \{t_1, t_2, \cdots, t_k\}$，其中 $t_i = \{i_{i1}, i_{i2}, \cdots, i_{ik}\}$。关联规则是形如 $A \Rightarrow B$ 的蕴涵式，这里 $A \subset I, B \subset I$，$A \cap B = \phi$。规则 $A \Rightarrow B$ 在实物 D 中成立，具有支持度 α，其中 α 是 D 中事物包含 $A \cup B$（即 A 和 B）二者的百分比，记为 $P(A \cup B)$。

定义 6.2 支持度（Support）

关联规则 $A \Rightarrow B$ 在事物集 D 中成立，具有支持度 α，支持度是事务同时包含 A 和 B 的百分比，记为 Supp（$A \cup B$），即

$$\text{Supp}(A \Rightarrow B) = \text{Supp}(A \cup B) = P(A \cup B) \qquad \text{（公式 6-1）}$$

支持度的作用是反映了 A 和 B 中所含的项在事务集合中同时出现的概率。例如，支持度为 40% 表示“在所有顾客中同时购买啤酒和尿布的顾客的概率为 40%”。

定义 6.3 置信度（Confidence）

关联规则 $A \Rightarrow B$ 在事务集 D 中具有置信度 β，置信度是事务集 D 中包含 A 事务同时也包含事务 B 的百分比，记为 Conf（$A \cup B$），即

$$\text{Conf}(A \Rightarrow B) = P(B|A) = \frac{\text{support} - \text{count}(A \cup B)}{\text{support} - \text{count}(A)} \qquad \text{（公式 6-2）}$$

其中，support - count($A \cup B$) 是包含项集（$A \cup B$）的事务数，support - count(A) 是包含项集（A）的事务数。例如“在所有购买啤酒的顾客中有 70% 的顾客购买了尿布”，则其置信度为 70%。

定义 6.4 项集与项集的频率

项的集合称为项集（Item - set），包含数据项数目为 k 的项集称为 k - 项集。项集的出现频率是包含项集的事务数，简称为项集的频率、支持率或计数。

定义 6.5 频繁项集

如果项集出现的频率大于或等于最小支持度，即满足最小支持度阈值，则称它为频繁项集（Frequent Item - set）。频繁 k - 项集的集合通常记为 L_k。

定义 6.6 强规则

同时满足最小支持度阈值（min - sup）和最小置信度阈值（min - conf）的规则称作强规则。关联规则挖掘的任务就是在事务集 D 中找出满足用户给定的最小支持度 min - sup 和最小置信度 min - conf 的强关联规则。强关联规则对应的项集必定是频繁项集 $A \cup B$，而频繁项目集 $A \cup B$ 导出的关联规则 $A \Rightarrow B$ 的置信度又可由频繁项集 A 和 $A \cup B$ 的支持度计算。因此，关联规则挖掘可分解为两个步骤：

（1）找出存在于数据库中的所有频繁项集，即找出那些支持度不小于事先给定的支持度阈值的项集。

（2）在找出频繁项集的基础上产生强关联规则，即产生那些支持度和置信度分别大于或等于事先给定的支持度阈值和置信度阈值的关联规则。第一个步骤的任务是迅速高效地找出 D 中的全部频繁项目集，是关联规则挖掘的中心问题，是衡量关联规则挖掘算法效率

的主要指标，目前大多数有关关联规则挖掘问题的研究都是针对第一个子问题而展开的。

第二个步骤的算法相对简单，它只是由频繁项集产生强规则的枚举探查过程。具体实现方法为：

①对频繁项集 L 中的每一个元素 l（即 l 是 L 中的某个频繁项集），产生其所有非空子集；

②对于 l 的么一个非空子集 s，如 $\text{support}(l)/\text{support}(s) \geqslant \min - \text{conf}$，则输出强规则“$s \Rightarrow (l - s)$”。

综上所述，关联规则挖掘的基本模型可用图 6 - 1 表示，其中 D 为数据集，Algorithm - 1 为频繁项集的搜索算法，Algorithm - 2 为关联规则的产生算法，R 为挖掘出的关联规则集合。用户通过指定最小支持度阀值（min - sup）和最小置信度阀值（min - conf）分别于算法 Algorithm - 1 和 Algorithm - 2 交互，并通过与 R 的交互，来对挖掘结果进行解释和评价。

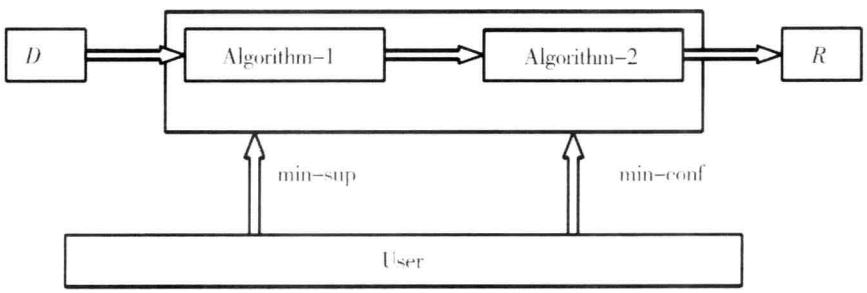


图 6 - 1　关键规则挖掘的基本模型图

6.2　APRIORI 算法

Apriori 算法使用一种称作逐层搜索的迭代方法。k 项集用于搜索（$k+1$）项集。首先找出频繁 l 项集的集合。该集合记作 L_1，L_2 的作用是找频繁 2 项集的集合 L_2，而 L_2 的作用是找 L_3，如此下去，直到不能找到频繁 k 项集。为了找到每个 L_k，需要进行一次数据库扫描。为提高频繁项集逐层产生的效率，Apriori 性质被用于压缩搜索空间，那就是频繁项集的所有非空子集也是频繁的。该性质基于如下观察：假如项集 I 不是频繁的，那么项 A 添加到 I，则结果集 $I \cup A$ 不可能比 I 更频繁出现。因此，$I \cup A$ 也不是频繁的。

利用这一性质我们看看如何由 L_{k-1} 找 L_k。这是分连接和剪枝两个步骤进行的。

（1）连接步：通过 C_k，连接产生候选项集 L_k，从而找到 L_k。设 $L_{k-1}[j]$ 为 L_{k-1} 的第 j 项，假设事务中的项按字典顺序排列，并设 l_1 与 l_2 为 L_{k-1} 中的项集且有相同的前 $k-2$ 项集，则 l_1 与 l_2 产生的候选 k - 项集为 $l_1(1), l_1(2), \cdots, l_1(k-1), l_2(k-1)$。由于 C_k 规模可能很大，可以利用 Apriori 算法的性质“任何非频繁项集 $k-1$ 项集的超集不可能是频繁项集”，实现减枝压缩 C_k。

（2）剪枝步：由于候选项集 C_k 是频繁项集 L_k 的超集，也就是说候选项集的成员中可以有非频繁项集，但所有的频繁 k 项集都包含在 C_k 中。为验证候选项集的频繁性，需对数据库进行扫描，确定每一候选项集的支持数，然后根据最小支持度 min - sup 确定频繁项

集 L_k。然而 C_k 可能很大，这样所涉及的计算量就很大。为压缩 C_k，可以使用 Apriori 性质，即任何非频繁的 $k-1$ 项集都表示频繁 k 项集的子集。因此，如果候选 k 项集的 $k-1$ 项子集不在 L_{k-1} 中，则该候选项集也不可能是频繁的，从而可以从 C_k 中删除。

具体的 Apriori 算法如下：

①Apriori 算法的频繁项集产生

输入：事务数据库 D，最小支持度 min - sup；

输出：D 中的频繁项集 L

步骤：

```
k = 1;
L_k = find - frequent - 1 - Item - set (D);    //发现所有频繁 1 项集
For (k = 2; L_{k-1} ≠ φ, k + +)
{
C_k = Apriori - gen (L_{k-1});    //产生候选项集
for 每个事务 t ∈ D
{
C_t = subset (C_k, t);    //识别属于 t 的所有候选
for 每个候选集 c ∈ C_t
c. count + +;    //支持度技术增值
    }
L_k = {c | c ∈ C_t ∧ c. count ≥ N × min - sup}    //提取频繁 k 项集
    }
Return  L + ∪_k L_k;
Apriori - gen 算法步骤：
Apriori - gen (L_{l-1}: frequent (k - 1) - Itemsets)
for each Item - set  l_1 ∈ L_{k-i}
for each Item - set  l_2 ∈ L_{k-i};
if (l_1 [1] = l_2 [1]) ∧ (l_1 [2] = l_2 [2]) ∧ ⋯ ∧ (l_1 [k-2] = l_2 [k-2]) ∧ (l_1 [k-1] < l_2 [k-1]) then
{    c = l_1 ∞_12;
if has - infrequent - subset (c, L_{k-1})    then
delete c;
else
add c to  C_k;
    }
Return  C_k;
has - infrequent - subset 算法步骤：
for each (k - 1) - subset s of c
if  s ∉ L_{k-1}  then
return TRUE;
return FALSE;
```

②由频繁项集产生关联规则

一旦从数据库 D 中的事务找出频繁项集，由它们产生强关联规则是直截了当的。因为关联规则挖掘的两个步骤中，第一步找出所有频繁项集是最难的，挖掘关联规则的总体性能由这步决定；第二步比较容易，对于置信度，可以用公式 6-2 进行计算。因此，关联规则可以由如下步骤产生：对于每个频繁的项集 l，l 的每一个非空子集 s，如果 support-count(l)/support-count(s) ≥ min-conf 则输出规则 "$s \Rightarrow (l-s)$"。其中，min-conf 是最小置信度阀值。由于规则由频繁项集产生，每个规则都自动满足最小支持度。

为了能够清晰地阐述 Apriori 算法的挖掘过程，举例如下：

设事务数据库 D 如表 6-1 所示，D 中包含 9 个事务，即 $|D| = 9$，最小的支持数 mincount = 2，即最小的支持度 minsup = 2/9 = 22%。挖掘频繁项目集的具体过程如下所述：

表 6-1　事务数据库 D

Tid	Itemset
T1	A, B, E
T2	B, D
T3	B, C
T4	A, B, D
T5	A, C
T6	B, C
T7	A, C
T8	A, B, C, E
T9	A, B, C

第一步，通过对算法的第一次迭代，对事务数据库进行一次扫描，计算出 D 中所包含的每个项目所出现的次数，然后生成候选 1-项集的集合 C_1。

第二步，在已设定的最小支持度的要求下，从 C_1 中可以确定出频繁 1-项集，即 L_1。鉴于此时所有的项目全部满足 mincount 的要求，因此 L_1 与 C_1 是等价的。

第三步，为了产生频繁 2-项集，执行 Apriori-gen 中的第七步（即 $L_1 \triangleright\triangleleft L_1$），生成候选 2-项集的集合 C_2，然后扫描事务数据库对 C_2 中的项目集进行计数。

第四步，根据 mincount 从 C_2 中确定 L_2，也就是把 C_2 当中满足 mincount 要求的首选项目集放入 L_2 当中。

第五步，为了产生频繁 3-项集，因此执行 Apriori-gen 中的第七步（即 $L_2 \triangleright\triangleleft L_2$），生成候选 3-项集的集合 $C_3 = \{\{A,B,C\},\{A,B,C\},\{B,C,D\},\{B,C,E\},\{B,D,E\}\}$。由于 $\{B,C,D\},\{B,C,E\},\{B,D,E\}$ 的子集中包含有非频繁的 2-项集，所以根据 Apriori-gen 中的剪枝步骤，它们剪掉，然后扫描事务数据库对其中的 C_3 的项目集进行计数。

第六步，根据 mincount 从 C_3 中确定出 L_3，即把 C_3 中满足 mincount 要求的候选项目集放入 L_3 当中。

第七步，为了产生频繁 4-项集，因此要执行 $L_3 \triangleright\triangleleft L_3$，生成候选 4-项集的集合 $C_4 = \{\{A,B,C,E\}\}$。由于 $\{A,B,C,E\}$ 的子集中包含有非频繁的 3-项集 $\{B,C,E\}$，因此

C_4 将其剪掉。此时即为空，Apriori 算法的整个执行过程结束，如图 6 - 2 所示。

在找到了事务数据库中的所有的频繁项集后，可以利用这些频繁项集产生相应的关联规则，产生关联规则的步骤如下：

①第一个步骤是产生所有频繁 1 - 项集的非空子集。

②对于 l 的每个非空子集 m，如果 $\text{support}(l)/\text{supprot}(m) \geqslant \text{min}cof$，则输出规则“$m \rightarrow (l - m)$”。

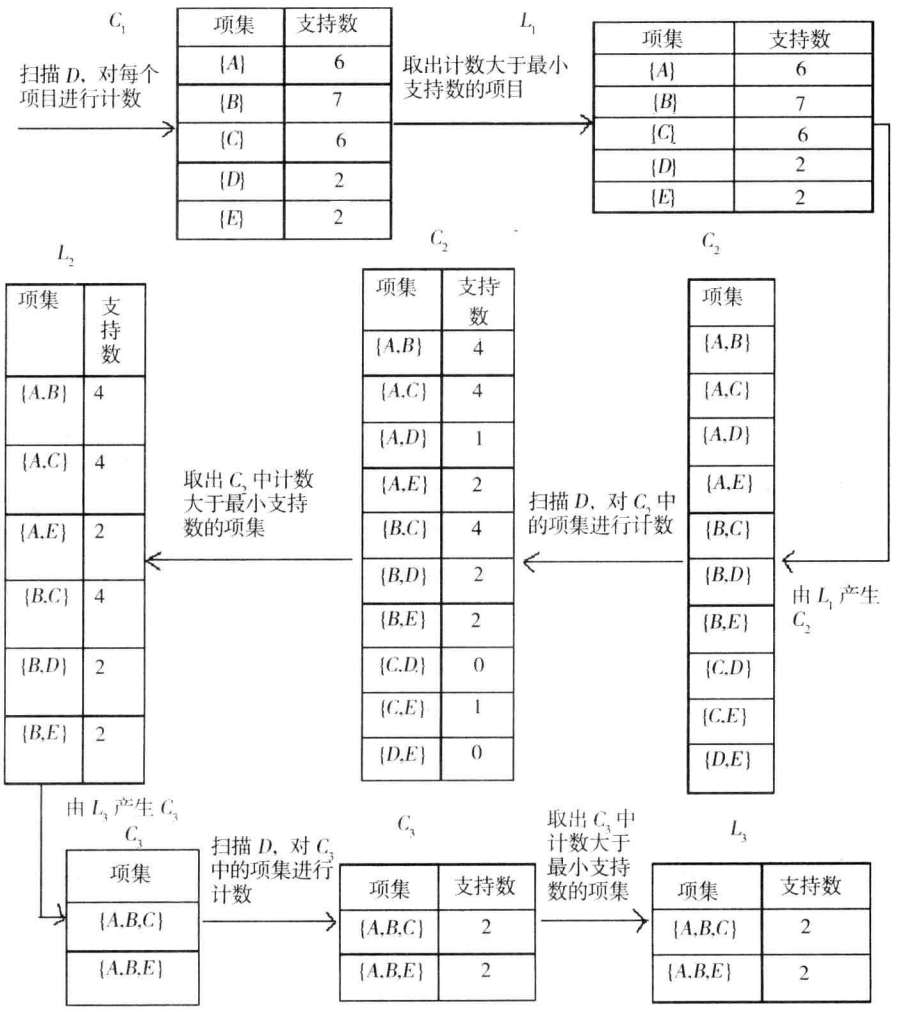


图 6 - 2　Apriori 算法执行过程

如上面的例子当中产生的频繁项目集 $l = \{A,B,E\}$，l 的非空子集有 $\{A,B\}$、$\{A,E\}$、$\{B,E\}$、$\{A\}$、$\{B\}$、$\{E\}$，则运用以上的关联规则方法可以得到如下的关联规则：

$A \wedge B \Rightarrow E$　　confidence = (2/9)/(4/9) = 0.5

$A \wedge E \Rightarrow B$　　confidence = (2/9)/(2/9) = 1

$B \wedge E \Rightarrow A$　　confidence = (2/9)/(2/9) = 1

$A \Rightarrow B \wedge E$　　confidence = (2/9)/(6/9) = 0.33

$B \Rightarrow A \wedge E$　　confidence = (2/9)/(7/9) = 0.29

$E \Rightarrow A \wedge B$　　confidence = (2/9)/(2/9) = 1

由以上的分析可以看出，Apriori 算法利用检查候选产生这个方法可以大幅度地压缩候选集的大小，得到令人满意的结果，但是这个算法当中依然是存在着几个不容忽略的情况：

（1）此算法在计算的过程当中需要产生大量的候选项目集。所以当候选集的基数很大，如频繁 1 - 项集很大的时候，那么由此而产生的候选集就会以指数的方式增长，这就会给 I/O 带来巨大的计算负荷，最终也会造成算法的执行效率大打折扣。例如，如果一个项目的频繁 1 - 项集有 10^4 个时，那么它所产生的候选 2 - 项集的数量也许会多达 10^7 个；那么如果一个长度为 100 的长频繁模式出现时，为此而付出的代价是产生 2^{100} 个候选项目集，由此可以看出对算法效率所产生的影响。

（2）此算法需要对数据库进行多次的扫描，产生候选集需要进行模式上的匹配，如果对大型数据库的数据进行处理，执行效率会大大地降低，局限性很大，只适合应用于小容量数据库。

（3）对稀有的信息无法进行分析。确定频繁项集是使用了最小支持度，所以，对于一些支持度计数小于最小支持度的项集无法进行分析，但是有些稀有的信息又是很有价值的。如果采用把最小支持度调小些，一方面减低了效率，另一方面又会生成很多无用的信息。

（4）只使用一个最小支持度，没有考虑各属性重要程度的不同。

（5）算法只是针对单维布尔关联规则的挖掘，对于其他性质的关联规则挖掘不适用。

由以上的五点可以得出，Apriori 算法使用于处理小数据量的数据库，对于大型的数据库和数据仓库，执行效率会大打折扣。从当今的发展形势来看，数据库中的数据量正在以指数的形式上涨，因此 Apriori 算法很难被运用于实际生活当中，因此对于 Apriori 算法的改进成为了很多研究关联规则的学者们研究的热点。

6.3　从频繁项集到强关联规则

支持度和置信度是描述关联规则的两个重要概念，支持度用于衡量关联规则在整个数据集中的统计重要性，后者用于衡量关联规则的可信程度。所有支持度大于最小支持度的项集称为频繁项集，关联规则由频繁项集产生。同时满足最小支持度闭值和最小置信度闭值的规则称作强规则，约束规则是在强规则的基础上提取的符合条件的规则。一般来说，只有支持度和置信度均较高的关联规则才可能是用户感兴趣的、有用的关联规则。由关联规则挖掘算法挖掘得到的关联规则也是由最小支持度和最小置信度决定的。由此我们需要考虑一个问题，如何设定合理的支持度和置信度？以下先看一个有关支持度置信度的例子。

对于某条规则，如 $(A \Rightarrow \alpha) \Rightarrow (B \Rightarrow b)$（support = 30%，confidence = 60%）。其中 support = 30% 就是意味着，在所有的数据记录中同时出现 $A = a$ 和 $B = b$ 的概率。而 confidence = 60%，即所有数据记录中，在出现 $A = a$ 的情况下出现 $B = b$ 的概率，即条件概率。支持度揭示了 $A = a$ 和 $B = b$ 同时出现的频率，如果 $A = a$ 和 $B = b$ 一起出现的频率

非常小，那么就说明了 $A = a$ 和 $B = b$ 之间的联系并不是很大，我们也就没有必要在其身上放太多的心思；但若 $A = a$ 和 $B = b$ 一起出现的频率非常高，那么它们总是相互关联，这是我们可理解的。置信度揭示了当 $A = a$ 出现时，$B = b$ 是否一定会出现，如果出现则其概率是多大。如果置信度为100%，则说明当 $A = a$ 出现时，$B = b$ 一定会出现，那么我们在研究其中一个时，就没有理由不考虑另一个对其的影响。如果置信度太低，那么就会产生这样的疑问，$A = a$ 和 $B = b$ 的关系并不大，也许 $A = a$ 关联的并不是 $B = b$。

如果关联规则挖掘依赖于一个固定的支持度和置信度闭值，其缺点主要有：

①如果支持度和置信度闭值设置得过高，虽然能减少挖掘时间，但是容易造成一些隐含于数据中非频繁特征项或稀少特征项中的规则被忽略掉，难以发现足够有用的规则。

②如果为了发现所有的规则，而把支持度和置信度设得过低，又可能会产生过多的规则，甚至产生大量冗余和无效的规则，同时由于算法存在的固有问题，会导致高负荷的计算量，大大增加挖掘的时间。

目前对于支持度和置信度值的设定并没有很好的方法，也少有关于这方面的研究。现有几种看法：①由业务专业人员根据经验设定；②设定支持度和置信度上确界和下确界；③建立支持度和置信度有关数据集的函数。

无论采用哪种方法，若采用一个固定的支持度和置信度闭值，难免会缺乏灵活性，所需规则的确定也过于简单。若能根据需要变动支持度和置信度闭值产生有所不同的关联规则，将对应用关联规则有所帮助。先通过最小支持度闭值产生最多频繁项集，然后通过设定的不同的支持度和置信度闭值产生有所不同的强关联规则，再用排序规则组对数据进行填补，最后通过数学统计分析选择合理的填补值。这样选择填补值既能充分应用关联规则挖掘发现变量关系的作用，也能反映填补值的不确定性。

6.4 提高 APRIORI 算法的效率

6.4.1 国内算法改进研究现状

在原有的关联规则 Apriori 算法中，它自身也存在着改进的方面，但是由于其自身算法的局限性，在其实际的应用过程当中，还是有很多问题需要去改进和优化，正是在这个基础上，有些学者相继提出了相关的优化方法，目前国内对于 Apriori 算法的改进也有大量发表的文章，主要分为以下几类改进：

（1）基于新的理论基础：尚学群从并行的角度对关联规则挖掘算法进行了改进，介绍了当前并行关联规则挖掘方面的研究发展，对一些典型算法进行了分析和评价，从并行度、负载平衡以及数据库的集成等方面展望了并行关联规则挖掘的研究方法；陆琳等提出了一种新的基于混合蚁群算法的关联规则挖掘方法，利用遗传算法对蚁群算法的参数组合进行优化，最终确定出一组较为理想的参数组合，提高了算法的运算性能；贺超波等提出一种以粗糙集理论为基础的关联规则挖掘方法，该方法首先利用粗糙集的特征属性约简算法进行属性约简，然后在构建约简决策表的基础上应用改进的 Apriori 算法进行关联规则挖掘。

（2）基于已有改进算法：秦吉胜针对 AprioriHybird 算法的瓶颈提出了一种使用支持度

矩阵对频繁 2 - 项集快速验证的方法，并给出了一种简单易行而又高效地逐步缩减交易数据库的方法，加快了对候选频繁 k 项集的验证速度；范明对 FR - growth 进行了改进，通过引入被约束子树，算法在挖掘频繁模式时不生成条件 FP - 树，从而大大提高了频繁模式挖掘的时空效率；杨明等对 FP - tree 进行了改进，引入了前缀广义表 - PGList，并提出了基于 PGList 的关联规则挖掘（MARBPGL）与增量式更新算法（IUABPGL）。

（3）基于 Apriori 算法需产生大量候选集的缺点：朱烨等针对 Apriori 算法存在的问题，提出新的算法 New - Apriori，该算法改变由低维频繁项目集到高维频繁项目集的多次连接运算，直接从 1 - 频繁项目集产生高维频繁项目集，在某一定程度上提高了运算效率；陈文庆等提出利用 Hash 表存储技术对该算法进行改进，通过删除项 Hash 表来减少生成候选集的数量，从而提高算法的效率。

（4）基于 Apriori 算法需多次扫描数据库的缺点：赵松等针对 Apriori 算法需多次扫描事务数据库的问题，在候选集 C_1 中增加事务标识符列表 Tid_ list 实现只需扫描一遍事务数据库，提高了算法效率；郭有强等提出了一种渐进式挖掘算法，算法充分利用以往过程中的信息，无需再次扫描基础数据集，对新增数据集部分也只扫描一次，能够有效地更新挖掘所需的时间成本；李超等对基于矩阵的 Apriori 算法进行了改进，同时改进了发现关联规则算法，将 Apriori 算法的剪枝与矩阵联系起来，可以大大减少扫描数据库的次数，从而提高算法的效率，在生成关联规则中，利用了概率论的基本性质也大大减少了计算量。

（5）从减少扫描项集角度考虑：张月琴等讨论粒计算在关联规则挖掘中的应用，通过对基本信息粒的划分、对粒子对象集合的映射，减少扫描项集所在的对象集合，提高算法的运行效率，从而更好地处理海量数据的规则发现，更适合于支持度较小、复杂度较高的数据集。

6.4.2 改进算法

在改善 Apriori 算法效率的各个研究中，主要朝着降低所需的计算量与减少扫描数据库的次数的方面来进行，接下来介绍几个 Apriori 的改进算法。

（1）Partition 算法

Savasere 于 1995 年提出了 Partition 算法，其主要的概念是由 Apriori 算法延伸而来，在搜索频繁项集以及挖掘关联规则等方面，其做法仍与 Apriori 算法大同小异。但如同上一节所述，在挖掘关联规则时，Apriori 算法将花费许多的时间在数据库的搜索上，这是一个很大的负担，而 Partition 算法所要改进的地方便是减少此搜索数据库所花费的成本，进而大大地提升其整体效能。Partition 算法先将数据库分成许多段，它的核心思想是：整个数据库上的频繁项集至少在数据库的一个分段上是频繁的。用反证法可以轻而易举地证明这点，那么，每个分段上的频繁项集集合的并集就是整个数据库上潜在的频繁项集的集合。

整个算法可以分为两个步骤：（1）将数据库分为许多小段，每个小段都能完全装入内存，这样就能调用任何一种频繁项集的挖掘算法在这个分段上挖掘这个分段的频繁项集集合。这只需要扫描一次数据库。（2）将每个分段的频繁项集合并起来，结果就是整个数据库上的一个巨大的候选集。现在需要对数据库进行第二趟扫描来验证每个候选项集是否是真正的频繁项集。

在第一个步骤中，Partition 算法将各个分段当成是一个小型的数据库，其大小可容纳

于内存中。而后针对各个分段，分别去计算其中的相关项集的支持度以找出各个分段内的频繁项集，因此必须加以存储，用于后来判断真正频繁项集之用。

接下来在第二个步骤中，根据在第一个步骤中所找出的所有分段的频繁项集，来找出在全部数据库中真正的频繁项集。因为如果一个项集在各个分段内都是非频繁的，则其在整个数据库中也必为非频繁的；而如果在某一分段中为频繁的，则其便有可能是一个真正的频繁项集。利用此特点，将各个分段中所得的分段性频繁项集集合起来，并再次搜索数据库以计算其支持度，便可以得到所有真正的频繁项集。

利用 Partition 算法只需要搜索整个数据库两次，所以可大大降低其 I/O 的成本，但是会造成下列问题：

①在第一个步骤中所产生的频繁项集只能算是分段中的频繁项集，并不一定表示在整个数据库中也是频繁项集，因此可能会产生过多的频繁项集，导致第二次扫描数据库做确认时效能不好。

②为了减少频繁项集在不同的分段中重复产生，而且为了能利用估计的方式提早结束挖掘，在系统做挖掘之前必须先将数据库做排序，而对于数据库做排序的时间复杂度，虽然作者并未在其文章中描述，但其所耗费的时间不容忽视。

③由于分段式挖掘会对各个数据分段做挖掘，所以其计算量会比 Apriori 算法更大。

④分段后各区的阈值必然需要向下调整，使得原本不会为候选项集的组合也都成为候选项集，增加了许多工作量。

⑤将数据库分段，分段的数量与分段的方法并无一定的标准，可能导致每次挖掘的结果都会有不同。

（2）Sampling 算法

挖掘频繁项集需要花费很多时间的主要原因在于数据库中的数据量过于庞大，加上需要多次扫描数据库，如果能够有效地减少所需要处理的数据量便可增加挖掘的效率。Toivonen 在 1996 年提出 Sampling 算法。他希望利用抽样方式从原始数据库中抽取样本，以便直接存储在内存中，再针对样本数据库挖掘频繁项集，以减少挖掘时间。这是因为样本体积比原始数据库小了许多，所以可以能快速挖掘出频繁项集。采用抽样技术产生的样本数据必定含有抽样误差，所以 Toivonen 提出利用降低最小支持度以避免遗漏频繁项集。Toivonen 提出的抽样算法是利用单纯随机抽样法来进行抽样的动作，由于样本数据库所含的数据量较少，所以能借此针对样本数据进行挖掘以减少原本所需要耗费的时间，提高挖掘效率。单纯随机抽样法具有简单快速的特点，但由于单纯随机抽样法进行时，并不会对抽样方法做任何限制，所以比较容易发生数据扭曲，如果数据扭曲严重的话，会造成样本数据库不具有代表性，会导致用样本数据库挖掘出来的频繁项集与用原始数据库所挖掘出来的结果不符合。另外，在抽样算法中，降低最小支持度的方式虽然可以避免频繁项集的遗漏，但是会造成不应被看做频繁项集的候选项集，却被当做频繁项集的情况，此种情况将会对后续的关联规则挖掘带来严重影响。

（3）DHP 算法

在 Apriori 算法的推导过程中，都是从单一项目开始，先组合出所有可能的候选项集，再从这些候选项集中找出真正的频繁项集，而下一层的候选项集就是将上一层的频繁项集排列组合，再重新分析，往后就如此循环下去，所以当重复 n 层之后，就会出现 $n+1$ 层的频繁项集。但从 Apriori 算法发现，其中最大的问题就是每个候选项集都必须经过上一

层的频繁项集排列组合而成，再将所有的候选项集与数据库比对以计算其支持度，所以效能将会随着候选项集数量的增加而降低，往往在第二层的过程中就几乎消耗掉所有的执行成本，所以候选项集的数量就是影响效能的关键所在，减少候选项集就能减少比对的次数，所需要的时间就能减少。

Chenetal 提出的 DHP 算法，主要利用了删减不必要候选项集的概念来改善挖掘关联规则时的效率。DHP 算法以 Apriori 算法为基础，但它引入了 hash table 的结构，并根据统计学上的定理，将其转换为非频繁项集的删选机制，减少执行过程中不必要的项集的数量，降低所需要的计算成本，从而有效提升挖掘关联规则的效率。

DHP 算法的具体步骤为：（1）利用频繁 1 项集将第二层的候选项集建立 HASH；表（2）根据 HASH 表所产生的候选项集，筛选出频繁项集，再利用频繁项集削减数据库的大小；（3）产生频繁 2 项集之后，以后的步骤就与 Apriori 算法相同，直到无法再产生更高层次的频繁项集为止。

DHP 算法利用 hash table 的结构来删除大量不必要的候选 2 项集，虽然在一开始时其会花费额外的时间来建立 hash table，但对于其后的候选项集的产生却能有相当大的改善。总体而言，此方法的确能显著提高效能。

然而，要针对长度较长的项集设计适当的 hash table 是相当不容易的，一方面，在对应项集至 hash table 的过程中也可能牵涉繁杂的数学计算而加重执行负担；另一方面，hash table 中所累计的统计值是对项集的支持度的大约估算，仍需要扫描数据库以获取各个项集真正的支持度，所以在执行过程中所需扫描数据库的次数基本上是与 Apriori 相同的，上述两点是 DHP 算法仍需改进的地方。

（4）DIC 算法

Brinetal 在 1997 年提出了 DIC 算法，此算法以 Apriori 算法为基础加以改进。如前所述，在 Apriori 算法中，主要花费的时间成本在于数据库的搜索上，而 DIC 算法的设计理念也是在于设法将此 I/O 存取所花费的时间加以减少。DIC 算法与 Apriori 算法不同的地方在于其将数据库分割成许多个区段，在搜索数据库时是以区段为单位。当候选项集产生后需过滤出频繁项集时，DIC 算法可以不用完全搜索完整个数据库一次，因为在搜索的过程中，有些候选项集在若干区段中的支持度计数便已经大于或等于使用者所设定的最小支持度限制，所以在搜索完这些若干个区段之后便可以立刻求出频繁项集并直接连接产生下一层的候选项集，而不一定要等到整个数据库都搜索完为止。利用该特性，一旦新的候选项集产生后，便可直接在下一个区段开始去计算其支持度，而不必等到下一次重新搜索数据库时再由第一个区段开始去累计其支持度。然后在每个区段中反复执行，直到找出所有的频繁项集为止，所以可大大减少搜索数据库的次数而提升其效能。在最差的情况下，其搜索数据库的次数最多和 Apriori 算法所需要的次数一样。

DIC 算法利用区段的特性来达到动态产生候选项集以及计算其支持度的目的，但区段的大小会大大影响其整体效能。如果区段切割太大，则无法有效减少搜索数据库的成本；但如果区段切割太小，则又必须花费额外的成本在数据的记录以及计算上，因此区段切割太大或太小都不能达到一个良好的效能。

此外，虽然 DIC 算法与 Partition 算法都是利用将整个数据库切割成许多个区段的特性来达到其减少搜索数据库的目的，但这两个方法仍是有相当大的不同之处。在切割区段时，Partition 算法必须考虑内存的大小，使其区段能容纳在内存中，也因此其总共只需要

两次搜索数据库；DIC 算法则无此限制，其区段的大小可随意切割，但每个区段的大小却必须相同，另外其搜索数据库的次数也会随着情况而有所不同。这两个方法的另一个不同点在于，Partition 算法虽然可以减少搜索数据库的次数，但会产生过多的非频繁项集；而 DIC 算法不但可以减少搜索数据库的次数，且可以减少非频繁项集的产生。

（5）FP－growth 算法

频繁模式增长（Frequent Pattern growth），简称 FP－增长，它由 Han. Pei 和 Yin 于 2000 年提出，能够在不产生候选项集的情况下产生所有的频繁项集。频繁模式增长算法采用了一个两步骤分而治之的策略：第一遍扫描首先将数据库压缩为一棵频繁模式树（FP－tree），同时保留项集的相关信息；然后将压缩后的数据库分成一组条件数据库，每个条件数据库关联一个频繁项，再分别对它们进行挖掘。

FP－growth 算法成功地使用了新的数据结构 FP－Tree，避免了产生候选频繁项集。虽然 FP－growth 算法和 Apriori 算法之间差异很大，但它们的思路都是将频繁项集按一定的规律进行分类，分别对各类进行挖掘。Apriori 算法的分类依据是频繁项集的长度，按长度将频繁项集分类为 1－itemset，2－itemset，…，k－itemset。按长度顺序首先挖掘 l－itemset，再在 1－itemset 的基础上挖掘 2－itemset，一直下去，直到 k－itemset。FP－growth 算法将所有的频繁项按照支持度递减排序，从而得到序 F－list，任意一个事务中的频繁项按此顺序进行排序后插入 FP－tree 中。而 FP－growth 算法自下而上生成后缀项集，依据后缀项集构造条件 FP－tree，在其上递归挖掘。递归操作本身就是对条件模式基的进一步划分。但是 FP－growth 有以下几点不足之处：

①FP－growth 算法构造新 FP－tree，并将数据存放在内存中直到递归返回。如果每条事务都很长，则需要较深的递归，这样就会大量消耗内存。

②新条件 FP－tree 的构造需要进行条件频繁项支持度计数、排序、产生条件模式基、构造条件 FP－tree 等费时的操作，这将增加算法运算时间。

③FP－growth 算法将 FP－tree 区分为单路径（Single－path）和多路径（Multi－path），对于单路径，由树结点的所有组合方式生成频繁项集。但是采用这种方法必须每次判断 FP－tree 是否是单一路径，从而降低了程序效率。

（6）Pincer－Seareh 算法

Linetal 于 1998 年提出了 Pincer－search 算法，主要的目的在于当关联规则的长度相对较长时，能有效地将其关联规则挖掘出来。所谓的最大频繁项集，指的是此项集是频繁的，而其所有的子集也都为频繁的，且其无任何超集为频繁的。最大频繁项集并不一定只有一个，如果能快速地找出在数据库中所有的最大频繁项集，相对地便是找出了数据库中所有的频繁项集。

以往的算法在数据项目长度相对较短时能有不错的处理效率，但当数据库中的数据项目平均长度增长时，效率却剧烈降低，其原因在于可能的项集个数是随着数据项目的长度成指数成长。当数据项目的长度较长时，所产生的项集将相当庞大，这会耗费许多 CPU 的时间；另外，数据项目的长度较长也意味着将搜索更多次数据库，I/O 的时间也势必增加。因此，如果使用以往单纯由下而上逐层搜索的方式来处理较长的数据项目，则其花费的成本将随项目长度而大幅度增长。

所以 Pincer－Search 算法通过快速找出所有的最大频繁项集来解决以上问题。其主要的优点在于除了自下而上的搜索方式之外，另外还加入了由上而下的搜索机制。由上而下

的搜索方式是指先将所有物品项目组合成一个最大候选项集，而后根据频繁项集的所有子集合也必然为频繁的特性，利用已经找出的非频繁项集对最大候选频繁项集进行删减，以产生新的最大候选项集，因为最大候选项集如果包含一个非频繁项集，则其必定为非频繁的。利用这两种搜索机制同时向中心点搜索，便如同夹子逐渐夹紧般，当所有的最大频繁项集都找出后，意味着已经得知所有在数据库中的频繁项集，搜索便可以结束，因而可有效提升处理效率。虽然 Pincer - Search 算法对于较长数据项目能有不错的效能，然而其在搜索数据库方面仍是一次搜索整个数据库，会花费许多 I/O 成本，所以在这方面仍有改善的空间。

6.5 Web 挖掘

6.5.1 Web 挖掘任务

（1）挖掘出新的客户

在挖掘 Web 的客户访问数据信息的过程当中，利用相关的分类技术能够在 Internet 上挖掘出新的客户。商务网站的拥有者先通过对已访问者的信息根据其浏览和购买行为进行分类，并在此基础上分析已经购买商品的客户所具有的公共属性。在这些属性当中，找出一些关键的属性，通过这些关键属性可以决定他们的分类，并找出这些属性之间的关联关系。对于网站当中新的访问客户，根据刚刚已经总结出的属性联系，观察出新客户的一些关键属性，从而识别出新客户与老客户之间可能存在的一些属性交集，从而对这个新的客户进行归类划分。然后，根据上面得到的信息来判定这个客户到底是潜在的客户还是只是一个简单的访问者，并根据这个选择来决定是否把这个新的访问者作为一个潜在的客户对待。通过以上过程把这个新客户所具备的客户类型确定后，可以对客户动态地展示 Web 页面，而次页面所存在的内容主要在于客户和销售商所存在的关系。若已成为了潜在的客户，销售商就会对已承认的客户展示一些特殊的、个性化的网页内容。

（2）提高服务的个性化水平和质量，使客户在网站上可以停留更长的时间使电子商务与人们的实际生活越来越紧密，而且电子商务使得传统客户与零售商所存在的空间距离已经几乎为零。在 Internet 上，所有的销售商对于客户来说都是相同的，不存在什么差别，所以怎样使客户能在自己的网站上停留更长的时间，也就成为销售商最为关心的一个问题，也是现在许多销售商所面临的一个挑战。因此，为了能在这个竞争当中占得先机，首先必须要非常清楚客户以前所浏览的网页以及所关心的商品信息，以此来判断消费者的兴趣爱好和相关需求，并提供具有美观度的 Web 页面，以使客户能够更加满意，即使同样的商品和价格也依然会选择在你的网站上做更长时间的停留，并最终选择你的商品。伴随电子商务的不断发展，具备个性化的服务已是现在社会发展的趋势，通过个性化服务，可以加深用户的访问体验，提高电子商务网站在销售方面做到更加的全面和深入。数据挖掘技术在这个过程中的应用，通过对客户的访问行为、访问次数、访问所主要涉及的内容等信息进行有效地挖掘，提取相关客户的特征，获取客户在此次浏览过程当中所采取的访问模式和访问习惯，通过这样的方式识别用户的忠诚度、偏好度和满意度，深入掌握客户的兴趣和需求，在网站中主动向客户提供消费者感兴趣的商品，帮助消费者可以更加便捷地

找到相关商品，通过上述种种过程的进行，提高消费者对网站的满意度，使访问者最终转变成购买者。

（3）提高站点设计水平，提高站点的应用效率

数据挖掘技术可以很好地提高站点的利用效率，在这个过程中，Web 设计者不会只是完全依赖于专家的意见来设计自己的网站，而是根据消费者所具有的不同信息特征来整理和设计网站本身的结构和外在的界面。对于这种情况我们可以形象地用超市当中的货架来比喻网站本身的设计，也就是把具有一定关联度的相关商品摆放在同一个货架上，有助于消费者选择，从而增加销售量。而这样设计的目的，也就是为了方便顾客，给顾客留下一个好的印象，从而吸引顾客下次来光顾网站。

（4）降低运营成本，提高企业竞争力

通过挖掘商品访问和销售记录，销售商能够得到比较可靠的销售信息，获取消费者购物的喜好，确定顾客消费的生命周期，分析出顾客的将来行为，并根据顾客的行为提出有针对性的营销活动。通过对数据挖掘技术的合理利用，在分析过大量客户的消费行为后，可以设计出适合企业的最佳营销和策划方案，从而达到降低经营成本和提高企业竞争力的双重目的。

（5）提高市场预测的有效性

根据数据挖掘技术的相关理论，企业通过分析顾客潜在的商业行为，可以有效地估测出一定的市场投资所能带来的回报是多少，从而可以从市场中获取非常可靠的反馈信息。通过这些步骤的进行，不仅可以降低企业的运营成本，提高企业的营业额，还可以在以后的政策制定中占据主动地位。

（6）系统的安全性和各项性能都得以保障

通过在数据挖过程实施中的各种分析，特别是对奇异点的分析结果可知，通过这个结果对客户在购买或者消费过程中出现的一场情况可以做出准确而且迅速判断，从而有效防止客户流失，以及一些不法人员所进行的欺诈活动。对于通过非法途径进入网络，企图入侵网络的数据挖掘分析情况，可以通过这项技术帮助找到系统所存在的弱点并加以改进，从而提高了站点的安全防范系数，可以有效地防范黑客的攻击，保证电子商务在一个安全的环境下运行。

6.5.2 关联规则在 Web 挖掘中的应用实例

现在，通过互联网在电子商务网站上购物已经成为很多消费者的选择。本节中将以某一电子商务网站的购物记录为实例，利用优化后的 Apriori 算法进行数据挖掘。

首先，是电子商务网站的购物记录中的几个属性，商品订单（订单号，订单日期，商品号，商品名，用户 ID，用户名，商品价格，商品数量）。以某一个具体订单来做说明，如表 6－2 所示。

表 6－2　订单实例

订单号	订单日期	商品号	商品名	用户 ID	用户名	价格	数量
0800406003087	03/15/2009	1001	手机	004520	小飞侠	1200	1

在上面的属性中，存在着三个关系模式，商品订单（订单号，用户 ID，商品号，订单日期），商品信息（商品号，商品名，商品价格，商品数量），用户信息（用户 ID，用户名，用户电话），由于每个订单只有一个订单号，不同的商品有不同的商品号，用户也只有一个用户 ID，所以其关键字分别是订单号、商品号和用户 ID。通过订单号可以得到消费者购买的详细信息并确定消费者的喜好。

在这个事物数据库中，首先通过订单号确定了所购商品的信息，然后从这些得到的数据中，根据数据关联规则的数据挖掘方法找出几种的关联规则。其次，在通过订单号确定用户 ID 后可以得到用户信息，从而得到每个用户所购商品的信息。在下面的实例中主要是以商品信息来确定用户所购商品之间的关联性，即通过对所获得商品信息进行关联规则的数据挖掘来确定商品之间的关联规则。

这是某一电子商务网站中一时间段的购物记录，用关联规则数据挖掘的 Apriori 算法进行挖掘。该购物记录的事物数据库是 2009 年上半年的销售记录。下面的数据库表是事物数据库样式，如表 6 - 3 所示。在半年的购物记录的数据库中，有几百条的数据，在事物数据库中根据商品订单的模式中的属性通过关键字订单号的查询可以查找出多个商品号，然后根据商品模式中的属性，通过商品号可以得到销售商品的信息，从而可以确定每个订单号所对应的商品，也就是消费者的所购商品的信息，结果如表 6 - 4 所示。

然后，对这个生成的表单进行关联规则的数据挖掘，即用改进的关联规则 Apriori 算法进行数据挖掘，通过对数据库中半年数据量的数据挖掘，可以找出商品之间的关联规则，得到消费者对商品需求的关联规则，从而为决策者执行下一步的商业计划起到辅助决策的作用。

表 6 - 3　购物记录数据库

订单号	订单日期	商品号	商品名	用户 ID	用户名	价格	数量
0800406003001	02/01/2009	1001	手机	004520	小飞侠	1200	1
0800400603001	02/01/2009	1004	笔记本	004520	小飞侠	4500	1
0800406003003	02/01/2009	1005	数码相机	006540	互联网	2400	1
0800406003003	02/01/2009	2001	运动鞋	006540	互联网	350	1
0800406003003	02/01/2009	6021	护发素	006540	互联网	20	1
0800406003006	02/01/2009	1009	硬盘	003546	mike	300	1
0800406003006	02/01/2009	1015	摄像头	003546	mike	110	1
0800406003006	02/01/2009	4011	衬衣	003546	mike	150	1
0800406003006	02/01/2009	1011	U 盘	003546	mike	80	1
0800406003010	02/02/2009	2001	运动鞋	008653	六月雨	350	1
0800406003010	02/02/2009	5102	篮球	008653	六月雨	140	1
0800406003010	02/02/2009	2006	休闲皮鞋	008653	六月雨	320	1
……	……	……	……	……	……	……	……

表 6-4　生成的数据表单

订单号	订单日期	商品名
0800406003001	02/01/2009	手机、笔记本
0800406003003	02/01/2009	数码相机、运动鞋、护发素
0800406003006	02/01/2009	硬盘、摄像头、衬衣、U 盘
0800406003010	02/02/2009	运动鞋、篮球、休闲皮鞋
……	……	……

下面在事物数据库中提取出某个时段的具体例子加以说明，验证这个算法在实际中的应用。该验证数据是某电子商务网站的某时间段的销售记录。将最小支持度阈值设为 20%，最小置信度阈值为 70%，求取该事物数据库中的关联规则，如表 6-5 所示。下面给出了算法验证的过程，即改进算法的在实例中的应用步骤。将表 6-6 作为事务数据库，得表 6-7。利用改进的 Apriori 算法，分析事物数据库中的数据。

首先，找出事务数据库中的所有频繁项目集。

候选频繁项目集 1-项目集

C_1 = {休闲皮鞋，运动鞋，数码相机，润肤露，U 盘，手机，硬盘，笔记本，MP3，篮球，摄像头，牛仔裤，衬衣}

表 6-5　销售记录

订单号	订单日期	商品名
0800406003087	02/10/2009	数码相机、手机、休闲皮鞋、润肤露
0800406003090	02/10/2009	笔记本、摄像头、衬衣
2800406003096	02/10/2009	护发素、手机、硬盘、笔记本
0800406003102	02/10/2009	可口可乐
0800406003106	02/10/2009	牛仔裤、休闲皮鞋、润肤露
0800406003115	02/10/2009	休闲皮鞋、运动鞋、数码相机、润肤胡
0800606003040	02/10/2009	护发素、手机、衬衣
0800606003042	02/10/2009	休闲皮鞋、润肤露
0800606003054	02/10/2009	笔记本
0800606003056	02/10/2009	护发素、硬盘、笔记本、摄像头
0800606003066	02/10/2009	护发素、笔记本
0800606003072	02/10/2009	护发素、笔记本、衬衣
0800606003080	02/10/2009	护发素、笔记本
0800606003082	02/10/2009	护发素、硬盘、篮球
0800606003088	02/10/2009	运动鞋、篮球
0800606003100	02/10/2009	数码相机、护发素、笔记本、摄像头
0800606003114	02/10/2009	护发素、篮球、摄像头

表 6-6　事务数据库

TID	项目集
0800406003087	数码相机、手机、休闲皮鞋、润肤露
0800406003090	笔记本、摄像头、衬衣
0800406003096	护发素、手机、硬盘、笔记本
0800406003102	MP3
0800406003106	牛仔裤、休闲皮鞋、润肤露
0800406003115	休闲皮鞋、运动鞋、数码相机、润肤露
0800606003040	护发素、手机、衬衣
0800606003042	休闲皮鞋、润肤露
0800606003054	笔记本
0800606003056	护发素、硬盘、笔记本、摄像头
0800606003066	护发素、笔记本
0800606003072	护发素、笔记本、衬衣
0800606003080	护发素、笔记本
0800606003082	护发素、硬盘、篮球
0800606003088	运动鞋、篮球
0800606003100	数码相机、护发素、笔记本、摄像头
0800606003114	护发素、篮球、摄像头

表 6-7　事务数据库（C_1）的支持度阀值和置信度阀值

候选 1-项目集	支持度阀值	置信度阀值
休闲皮鞋	4	22
运动鞋	2	11
数码相机	4	22
润肤露	4	22
U 盘	9	50
手机	2	11
硬盘	3	17
笔记本	8	44
MP3	2	11
篮球	4	22
摄像头	3	17
牛仔裤	1	6
衬衣	3	17

根据最小支持度繁殖20%，得到频繁1-项目集 L_1 = {休闲皮鞋,数码相机,润肤露,U盘,笔记本,篮球}。

根据改进的Apriori算法中的Apriori-gen生成候选频繁项集，根据改进的Apriori算法中的Apriori-gen生成候选频繁项集 C_2 = {{休闲皮鞋、数码相机}、{休闲皮鞋、润肤露}、{休闲皮鞋、U盘}、{休闲皮鞋、笔记本}、{休闲皮鞋、篮球}、{数码相机、润肤露}、{数码相机、U盘}、{数码相机、笔记本}、{数码相机、篮球}、{润肤露、U盘}、{润肤露、笔记本}、{润肤露、U盘}、{U盘、笔记本}、{U盘、U盘}、{笔记本、U盘}}。

根据最小支持度阀值20%，得到频繁2-项目集 L_2 = {{护发素，润肤露}，{U盘，笔记本}}。

由上一步可得到频繁候选3-项集，C_3 = {{护发素、润肤露、U盘}、{护发素、润肤露、笔记本}、{护发素、U盘、笔记本}、{润肤露、U盘、笔记本}}。

同理可求频繁3-项目集为空，所以循环结束。

由 L_2 生成的关联规则如下：

对{护发素，润肤露}中的任意子集分别计算关联规则，{护发素} → {润肤露}，{润肤露} → {护发素}的置信度都为100%，同样{U盘，笔记本}中{U盘} → {笔记本}，{笔记本} → {U盘}的置信度分别为67%和75%。

根据最小置信度阈值为70%可得到最后的关联规则为{护发素} → {润肤露}、{润肤露} → {护发素}、{U盘} → {笔记本}，其余的都小于70%，应该删除。

表6-8为事物数据库（C_2）的支持度阈值和置信度阈值，候选2-项目集支持度阈值和置信度阈值。

表6-8　事务数据库（C_2）的支持度阀值和置信度阀值

候选2-项目集	支持度阀值	置信度阀值
护发素、硬盘	2	11%
护发素、润肤露	4	22%
护发素、U盘	0	0%
护发素、笔记本	0	0%
护发素、U盘	0	0%
硬盘、润肤露	2	11%
硬盘、U盘	1	6%
硬盘、笔记本	0	0%
硬盘、U盘	0	0%
润肤露、U盘	0	0%
润肤露、笔记本	0	0%
润肤露、U盘	0	0%
U盘、笔记本	6	33%
U盘、U盘	3	17%
笔记本、U盘	1	6%

由上述事例可以看出，该关联规则算法的优化，减少了对事物数据库的扫描次数，提高了原算法的效率。最终的计算结果反映了商品之间的关系，虽然其结果不一定是现实当中的因果关系，但是有些规则，例如｛笔记本｝→｛U 盘｝就有很高的信赖度，相当的合理而且是可以解释的。总之，通过上述事例可以知道，从事物数据中发现关联规则，并从这些关联规则中挖掘出有用的信息，对于改进商业活动的决策非常重要。

第7章 聚类分析

7.1 概述

“物以类聚，人以群分”，聚类是人类的一项最基本的认识活动。聚类的用途是非常广泛的。在生物学中，聚类可以辅助动、植物分类方面的研究，以及通过对基因数据的聚类，找出功能相似的基因；在地理信息系统中，聚类可以找出具有相似用途的区域，辅助石油开采；在商业上，聚类可以帮助市场分析人员对消费者的消费记录进行分析，从而概括出每一类消费者的消费模式，实现消费群体的区分。

聚类就是将数据对象分组成为多个类或簇（Cluster），划分的原则是使同一个簇中的对象之间具有较高的相似度，而不同簇中的对象差别较大。与分类不同的是，聚类操作中要划分的类是事先未知的，类的形成完全是数据驱动的，属于一种无指导的学习方法。

聚类分析源于许多研究领域，包括数据挖掘、统计学、机器学习、模式识别等。它是数据挖掘中的一个功能，但也能作为一个独立的工具来获得数据分布的情况，概括出每个簇的特点，或者集中注意力对特定的某些簇作进一步的分析。此外，聚类分析也可以作为其他分析算法（如关联规则、分类等）的预处理步骤，这些算法在生成的簇上进行处理。

聚类没有先验知识可以利用，是人类的一项最基本的活动，它是一种无监督的分类。但是聚类又有着与分类不同的概念与特点，二者的区别主要有以下几个方面：

首先，聚类是根据一定的聚类规则将拥有相同特征或者相似特征的数据聚在一起，它之前不需要依赖已经定义好的类或者带有类标记的实例，而是由聚类算法来确定样本标记。

其次，聚类与分类的区别在于，聚类涉及的数据集合的特征是未知的，并且在开始聚类之前，用户并不知道要把数据划分成几类，也不清楚分组的标准。在有些聚类算法中，如 K - Means 算法需要事先给出聚类的数目 k 值，而这个 k 值是凭用户的经验所得。分类是典型的有监督学习，分类的实例或者数据对象有类别标记，因此，用户之前是知道数据对象将会被分成几类的，然后运用分类器将要处理的数据分入不同的类别。

迄今为止，聚类还没有一个学术界公认的定义。这里给出 Everitt 在 1974 年关于聚类所下的定义：一个类（簇）内的实体是相似的，不同类（簇）的实体是不相似的；一个类（簇）是测试空间中点的会聚，同一类（簇）的任意两点之间的距离小于不同类（簇）的任意两个点间的距离；类（簇）可以描述为一个包含密度相对较高的点的多维空间的连通区域，他们借助包含密度相对较低的点的区域相分离。

事实上，聚类是一个无监督的分类，没有任何的先验知识可以用。若令 $X = \{x_1, x_2, \cdots, x_n\}$，则表示一个模式（实体、样本）的集合，$x_i$ 表示第 i 个模式，$i = 1,2,\cdots,n$，则聚类可描述为将 X 划分为 k 个类簇 $C_j, j = 1,2,\cdots,k$，即：

$$C_j \subseteq X, j = 1,2,\cdots,k$$

$$C_j = \{x_{j1}, x_{j2}, \cdots, x_{jne}\}$$

其中，ne 为第 j 个类簇中实体的个数。C_j 满足如下条件：

（1）$U_{j=1}^{k} C_j = X$；

（2）对于 $\forall C_m, C_n \subseteq X, C_m \neq C_n, C_m \cap C_n = \phi$（仅限于硬性聚类）。

在这里，我们可以用 proximity 函数来衡量实体之间的相似性距离，则任意两实体之间的相似性距离可用 proximity(x_y, x_s) 来表示，若满足：

$Y = \mathrm{MIN}_{\forall x_r \in C_m, \forall x_s \in C_n, \forall C_m, C_n \subseteq x \& C_m \neq C_n}(\mathrm{proximity}(x_r, x_s))$

$Z = \mathrm{MAX}_{\forall x_r, x_s \in C_m, \forall C_m \subseteq X}(\mathrm{proximity}(x_r, x_s))$

则恒有 $Y > Z$。

聚类主要包括以下几个过程：

（1）数据准备

数据准备包括特征标准化和降维。

（2）特征选择、提出

从最初的特征中选择最有效的特征，并将其存储于向量中。

（3）特征提取

通过对所选择的特征进行转换，形成新的突出特征。

（4）聚类（或分组）

首先选择合适特征类型的某种距离函数（或构造新的距离函数）进行接近程度的度量，然后执行聚类或分组。

（5）聚类结果评估

这是指对聚类结果进行评估。评估主要有 3 种：外部有效性评估、内部有效性评估和相关性测试评估。

7.2 数据结构与数据类型

7.2.1 数据结构

假设要聚类的数据集合包含 n 个数据独享，这些数据对象可能表示人、文档、房子、国家等。许多聚类算法选择如下两种具有代表性的数据结构：

（1）数据矩阵（Data Matrix）

这种数据结构又被称为“对象—属性”结构，类似于关系表的形式，它表示 n 个具有 p 个属性的对象，例如人可以用年龄、性别、审稿、体重、种族等属性来表示。这种数据结构是关系表的形式，或者直接看成 $n \times p$ 维的矩阵：

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

（2）差异度矩阵（Dissimilarity Matrix）

这种数据结构存储 n 个对象两两之间的近似性，表现形式是一个 $n \times n$ 维的矩阵：

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

在这里 $d(i, j)$ 是对象 i 和对象 j 之间相异性的量化表示，通常它是一个非负的数值。其中 $d(i, j) = d(j, i)$，当对象 i 和对象 j 越相似或“相近”，其值就越接近 0；两个对象差别越大，其值就越大。

数据矩阵经常被称为二模矩阵（two - mode），而相异度矩阵则被称为单模（one - mode）矩阵，这是因为前者的行和列代表不同的实体，后者的行和列代表相同的实体，许多聚类算法以相异度矩阵为基础。如果数据是用数据矩阵的形式表示的，那么在使用该类算法之前要将数据转化为相异度矩阵。

7.2.2 数据类型

聚类分析起源于统计学，传统的分析方法大多是在数值类型数据的基础上研究的。然而数据挖掘的对象复杂多样，要求聚类分析的方法不仅能够对属性为数值类型的数据进行，而且要适应数据类型的变化。一般而言，在数据挖掘中，对象属性经常出现的数据类型有区间标度变量，二元变量，标称型、序数型和比例标度型变量以及混合类型的变量。

数据挖掘的对象复杂多样，因此聚类分析中处理的数据类型也是多种多样的。总的来说，聚类分析中对象属性经常出现的数据类型有：区间标度变量，二元变量，标称型、序数型和比例标度型变量。

（1）区间标度变量

区间标度变量是一个粗略线性标度的连续度量。典型的例子则包括重量和高度，经度和纬度坐标，以及摄氏或华氏温度等。为了将数据或对象集合划分成不同类别，必须定义差异性或相似性的测度来度量同一类别之间数据的相似性和不属于同一类别数据的差异性。同时要考虑到数据的多个属性使用的是不同的度量单位，这些将直接影响到聚类分析的结果，所以在计算数据的相似性之前先要进行数据的标准化。

对于一个给定的有 n 个对象的 m 维属性数据集，可以利用以下标准化方法进行标准化：

平均绝对误差 S_p

$$S_p = \frac{1}{n} \sum_{i=1}^{n} |x_{ip} - m_p| \qquad \text{（公式 7 - 1）}$$

这里 x_{ip} 表示的是第 i 个数据对象在属性 p 上的取值，m_p 是属性 p 上的平均值。

$$m_p = \frac{1}{n}\sum_{i=1}^{n} x_{ip} \qquad \text{（公式 7-2）}$$

标准化度量 Z_{ip}：

$$Z_{ip} = \frac{x_{ip} - m_p}{S_p} \qquad \text{（公式 7-3）}$$

对于孤立点，平均绝对误差 S_p 比标准差 σ_p 具有更好的鲁棒性。在计算平均绝对偏差时，属性值与平均值的偏差 $|x_{ip} - m_p|$ 没有平方，因此孤立点的影响在一定程度上减小了。数据标准化处理后就可以进行属性值的相似性度量，通常是计算对象间的举例。对于 n 维向量 x_i 和 x_j，有以下几种距离函数：

①欧式距离：

$$D(x_i, x_j) = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \qquad \text{（公式 7-4）}$$

②曼哈顿距离：

$$D(x_i, x_j) = \sum_{k=1}^{n} |x_{ik} - x_{jk}| \qquad \text{（公式 7-5）}$$

③明氏距离：

$$D_m(x_i, x_j) = \left[\sum_{k=1}^{n} (x_{ik} - x_{jk})^m\right]^{\frac{1}{m}} \qquad \text{（公式 7-6）}$$

当 $m=2$ 时，明氏距离 D_2 即为欧式举例；当 $m=1$ 时，明氏举例 D_1 即为曼哈顿距离。对于欧式距离和曼哈顿距离满足以下条件：

a. $D(x_i, x_j) \geqslant 0$：距离是一个非负数值；

b. $D(x_i, x_j) = 0$：对象与自身的距离为零；

c. $D(x_i, x_j) = D(x_j, x_i)$：距离函数具有对称性；

d. $D(x_i, x_j) \leqslant D(x_i, x_k) + D(x_k, x_j)$：$x_i$ 到 x_j 的距离小于等于 x_i 到 x_k 和 x_k 到 x_j 的距离之和（三角不等式）。

（2）二元变量

二元变量只有两个状态：0 和 1。其中 1 代表相应的状态存在，0 代表所表示的状态不存在。二元变量又分为对称二元变量和非对称二元变量。前者是指变量的两个状态不具有优先权，两个状态的重要性一样，后者不同状态的重要性是不同的。对于对称二元变量，度量两个变量的差异度由简单匹配系数决定，而非对称二元变量的差异度则用 Jaccard 系数表示。

设两个对象 x_i 和 x_j，q 是属性值在两个对象中都为 1 的属性数目，s 是属性值在 x_i 中为 1 而在 x_j 中为 0 的属性数目，t 是属性值在两个对象中都为 0 的属性数目，p 表示所有属性的数目，$p = q + r + s + t$。

如果对象的属性是对称二值属性，则对象 x_i 和 x_j 的差异度计算采用简单匹配系数方法：

$$d(i,j) = \frac{r + s}{q + r + s + t} \qquad \text{（公式 7-7）}$$

如果对象的属性是非对称二值属性，若认为取 1 值的状态比取 0 值的状态重要，则对象 x_i 和 x_j 的差异度计算采用 Jaccard 相关系数方法：

$$d(i,j) = \frac{r+s}{q+r+s} \qquad \text{（公式 7-8）}$$

（3）标称型、序数型和比例标度型变量

①标称变量

标称变量是二元变量的推广，它可以有多个状态值，状态之间是无序的。具有这种数据类型的属性也称分类属性。它的差异度可用简单匹配法来计算：

$$d(i,j) = \frac{p-m}{p} \qquad \text{（公式 7-9）}$$

其中，m 是对象 x_i 和 x_j 取值相同的属性个数，而 p 是全部属性个数。

②序数型变量

序数型变量可以是离散的，也可以是连续的。在连续型的序数变量中，值的相对顺序是必要的，而其实际的大小则显得不重要。在相异度的计算中，需要把每个变量的值域映射到［0.0，1.0］上，以便每个变量都有相同的权重，然后采用距离的计算方法进行相异度的计算。

③比例标度型变量

比例标度型变量是非线性的标度取正的度量值，例如指数标度 Ae^{Bt} 或 Ae^{-Bt}，计算时可以采用与处理区间标度变量相同的方法。可以先进行对数变换，再进行计算；也可以借用序数性变量，再采用秩作为区间标度的值来对待。

④混合类型的变量

以上讨论了各种数据类型和它们差异度的计算方法。在实际数据库中，对象是由混合类型的变量描述的。在实际聚类分析中，有时将不同的类型属性组合在同一个差异度矩阵中进行计算。设数据包含 m 个不同类型的属性，对象 x_i 和 x_j 的差异度定义为：

$$d(x_i,x_j) = \frac{\sum_{p=1}^{m} \delta_{ij}^{(p)} d_{ij}^{(p)}}{\sum_{p=1}^{m} \delta_{ij}^{(p)}} \qquad \text{（公式 7-10）}$$

其中，如果 x_{ip} 或 x_{jp} 缺失，即 $x_{ip} = 0$ 或 $x_{jp} = 0$，且变量是不对称二元变量时，则指示项 $\delta_{ij}^{(p)} = 0$；否则 $\delta_{ij}^{(p)} = 1$；如果属性 p 是二元变量或标称变量：若 $x_{ip} = x_{jp}$，则 $d_{ij}^{(p)} = 0$，否则 $d_{ij}^{(p)} = 1$；如果属性 p 是序数型或比例标度型变量：将其转化为区间标度变量值对待。

7.3 聚类准则函数

在样本相似性度量的基础上，聚类分析还需要特定的准则函数才能把真正属于同一类的样本聚合到一个类中，而把不同类的样本分离开。聚类准则选得好，聚类质量就会高。聚类准则函数还可以用来评价聚类结果的质量，如果聚类质量不能满足要求，就要重复执行聚类过程，以便优化聚类结果。

常用的聚类准则函数有如下几种：

（1）误差平方和准则函数 J_c

这是最常用的一种聚类准则函数。假定样本集 $X = \{x_1, x_2, \cdots, x_n\}$，在某种相似性度

量基础上，它被聚类成 C 个分离开的类 $X_1, X_2, \cdots, X_c$，每个类分别包括 $n_1, n_2, \cdots, n_c$ 个样本。衡量聚类质量的误差平方和准则函数定义为：

$$J_c = \sum_{j=1}^{c} \sum_{k=1}^{nj} \| x_k - m_j \|^2 \qquad \text{（公式 7-11）}$$

其中，m_j 是类型 w_j 中样本的均值：

$$m_j = \frac{1}{n_j} \sum_{x \in X_j} x, j = 1, 2, \cdots, c \qquad \text{（公式 7-12）}$$

m_j 是 C 个聚类中心，可以用来代表 C 个类。

J_c 是样本和聚类中心的函数，在样本集 X 给定的情况下，J_c 的值取决于 C 个聚类中心。J_c 描述 n 个样本聚类成 C 个类型时所产生的总的误差平方和。J_c 值越大，说明误差越大，聚类效果越不好。因此，应该寻求使 J_c 最小的聚类结果，即在误差平方和准则下的最优结果。这种聚类通常称为最小方差划分。

误差平方和准则函数适用于各种类型样本比较密集且样本数目差别不大的样分布。当不同类型的样本数目相差较大时，采用误差平方和准则可能会把样本目多的类型分开，以便达到总的误差平方和最小，导致聚类结果不理想。

（2）加权平均平方距离和准则

$$J_j = \sum_{j=1}^{c} P_j S_j^* \qquad \text{（公式 7-13）}$$

其中，S_j^* 是类内样本间平均距离：

$$S_j^* = \frac{2}{n_j (n_j - 1)} \sum_{x \in X_i} \sum_{x' \in X_j} \| x - x' \|^2 \qquad \text{（公式 7-14）}$$

因此 J_j 是以先验概率 P_j 为加权的总类内平均平方距离之和，其中先验概率 P_j 可以用各类样本数 n_j 及样本总数 n 来估计，即：

$$P_j = \frac{n_j}{n}, j = 1, 2, \cdots, c \qquad \text{（公式 7-15）}$$

当各类样本数目差别比较大时，使用加权平均平方距离和准则比使用误差平方和准则更容易得到理想的聚类结果。

（3）类间距离和准则

为了描述聚类结果的类间距离的分布状态，可以使用类间距离和准则 J_{b1} 和加权的类间距离和准则 J_{b2}，它们的定义分别如下：

$$J_{b1} = \sum_{j=1}^{c} (m_j - m)^T (m_j - m) \qquad \text{（公式 7-16）}$$

$$J_{b2} = \sum_{j=1}^{c} P_j (m_j - m)^T (m_j - m) \qquad \text{（公式 7-17）}$$

其中，m_j、m、P_j 的定义如下：

$$m_j = \frac{1}{n_j} \sum_{x \in X_j} x, j = 1, 2, \cdots, c \qquad \text{（公式 7-18）}$$

$$m = \frac{1}{n} \sum_{k=1}^{n} x_k \qquad \text{（公式 7-19）}$$

$$P_j = \frac{n_j}{n} \qquad \text{（公式 7-20）}$$

类间距离和准则描述不同类型之间的分离程度，所以 J_b 的值越大，各类之间分离性越好，聚类质量越高。

7.4 主要聚类算法

目前文献中存在着大量的聚类算法，大体上，聚类分析算法主要分成如下几种，图7-1显示了一些主要的聚类算法的分类。

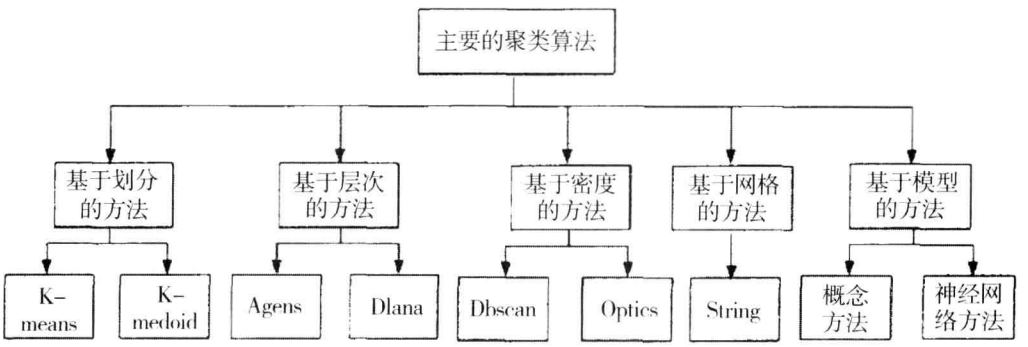


图7-1 聚类算法的分类图

7.4.1 基于划分的方法

划分聚类也叫分割聚类，对于一个给定的 n 个对象或元素的数据库，通过划分方法构建数据的 k 个划分，每个划分分别表示一个聚类，并且 $k \leqslant n$。也就是说，它将数据划分为 k 个组，而且满足如下要求：①每个组至少包含一个对象；②每个对象必须属于而且只属于一个组。对某些模糊划分技术，第二个要求可以放宽，即每个对象可以属于不同的组。划分方法首先要给定划分数目 k，同时创建一个初始划分，通常采用的方法是随机选取 k 个数据对象作为初始聚类中心点，然后采用一种迭代的重定位技术，尝试通过对象的划分和移动来改进划分。判断划分好坏的原则是：在同一个类中的对象之间应尽可能“接近”或相关，而不同类中的对象之间应尽可能“远离”或不同。

基于划分的聚类算法是一种基于爬山式的优化搜索算法，此法简单、快速而且有效，但是也存在不足之处，如：对初始值敏感；对输入顺序敏感；常陷入局部最优等等。根据对象在划分之间移动的衡量参数和簇的表示方法的不同，基于划分的聚类方法主要分为：K-means 算法，K-medoid 算法、Clarans 算法、Pam 算法等。

7.4.2 基于层次的方法

基于层次的方法对给定的数据对象集合进行层次分解，按数据分层建立组，形成一棵以组为节点的树。根据层次形成方式的不同，层次的方法可以分为分裂的方法和凝聚的方法。分裂的方法也称为自顶向下的方法，该方法一开始将所有的对象置于一个组中。在迭代的每一步中，一个组被分裂为更小的组，直到最终每个对象被分在单独的一个簇中，或者达到一个终止条件。凝聚的方法，也称自底向上的方法，该方法一开始将每个对象作为单独的一个组，然后把离得近的对象合并到一起，直到所有的组合并为一个（层次的最上

层），或者到达一个终止条件。层次聚类法与划分聚类法的区别在于，它并不试图寻找最优的聚类结果，而是按照一定的相似性判别标准，对最相似的部分进行合并。

层次聚类算法简单直接、易于理解和应用。它的缺陷在于，一旦一个步骤（合并或分裂）完成，它就不能被撤销，一旦某个划分错误，就无法改正。为改进层次方法的聚类质量，可以通过将层次聚类与其他聚类技术进行集成，形成多阶段聚类。基于层次的聚类算法有：Birch 算法、Cure 算法、Chameleon 算法等。

7.4.3 基于密度的方法

大多数划分方法都使用对象之间的距离来描述数据对象之间的相似程度，进而划分不同组实现聚类。这样的聚类方法对于发现球状簇的效果较好，而在发现任意形状的簇上就遇到了困难。专家学者们随之提出了基于密度的聚类方法，其主要思想是：只要邻近区域的密度（对象或数据点的数目）超过某一个阈值，就把它加到与之相近的聚类中去，继续聚类。

该类算法的优点是：领域知识独立；可以发现任意形状的类；对初始值、数据输入顺序不敏感；能够有效去除噪声。不足之处为：要对密度参数和噪声阈值进行仔细选择。典型的基于密度的聚类方法包括 Dbscan 算法、Optics 算法、Denclue 算法和 Subcluster 算法等。

7.4.4 基于网格的方法

基于网格的方法是采用了一个多分辨率的网格数据结构。它首先将数据空间划分成有限个单元 cell 的网格结构，所有的处理都是以单个的单元为对象的。这种方法的主要优点是处理速度很快，其处理时间与数据对象的数目无关，只与把数据空间划分为多少个单元有关。基于网格的聚类方法符合一个好的聚类算法的许多标准：能有效地处理大数据集；发现任意形状的簇；成功地处理孤立点；对输入顺序不敏感；不需要指定结果簇数目和领域半径等输入参数；而且能处理高维数据。代表算法有 Sting 算法、Clique 算法、Wave - Cluster 算法等。

7.4.5 基于模型的方法

基于模型的方法试图优化给定的数据和某些数学模型之间的适应性，这种方法通常是基于这样的假设：数据是根据潜在的概率分布生成的。基于模型的方法主要有两类：统计学方法和神经网络方法。概念聚类是一种统计学方法。对于给出的一组未标记的对象，它产生对象的一个分类模式，与传统的聚类方法不同，概念聚类除了确定相似对象的分组外，还向前走了一步，为每组对象发现了一个特征描述，即每组对象代表了一个概念或类。因此，概念聚类是一个两步过程，首先进行聚类，然后给出特征描述。在这里聚类质量不再只是单个对象函数，还加入了其他因素，如导出的概念描述的简单性和一般性等。

神经网络方法将每个类描述为一个标本。标本作为聚类的原型，不一定对应一个特定的数据实例或对象，根据某些距离度量，新的对象可以被分配给标本，被分配给一个类的对象的属性可以根据该类的标本的属性来预测。

7.4.6 模糊聚类

以上介绍的几种聚类算法可以给出确定的聚类结果，即每个数据点必须属于而且只能属于一个类。我们把这些聚类方法称为确定性聚类。然而现实中存在着大量界限并不分明的聚类问题，随着模糊集理论的提出，模糊聚类应运而生。在模糊聚类中，每个样本可以不再只属于某一类而是以一定的隶属度属于每一类。换句话说，通过模糊聚类分析，得到了样本属于各个类别的不确定性程度，即建立起了样本对于类别的不确定性描述，这样就更能准确地反映现实世界。

基于目标函数的模糊聚类方法首先由 Ruspini 提出，真正有效的模糊聚类算法是模糊 C 平均值算法（Fuzzy C - Means，FCM）。该算法由 Dunn 提出，Bezdek 将其进一步扩展，建立起了模糊聚类理论。从此，模糊聚类蓬勃发展，目前已经形成了庞大的体系。

7.5 K - means 聚类算法

K - means 算法是一种古老的、广泛使用的聚类算法。K - means 算法简单并且适用各种数据类型，而且有效。K - means 聚类算法的原理是：首先随机选取 k 个点作为初始聚类中心，然后计算各个样本到聚类中心的距离，把样本归到离它最近的那个聚类中心所在的簇；对调整后的新簇计算新的聚类中心，如果相邻两次的聚类中心没有任何变化，说明样本调整结束，这时某个误差平方和函数已经达到最小，聚类准则函数已经收敛。对于数据分布比较接近球状的情况有很好的聚类效果。K - means 算法是由 MacQueen 在 1967 年首次提出的，目前该算法已经成为数理统计、模式识别、机器学习和数据挖掘等领域应用最普遍的聚类算法之一，并衍生出多种变形算法，组成了 K - means 算法家族。这些 K - means类型的算法聚类速度快、易于实现，而且适用于文本、图像特征等多种数据的聚类分析。

7.5.1 K - means 算法流程

将 d 维数据集 $X = \{x_i \mid x_j \in R^d, i = 1,2,\cdots,N\}$ 聚集成 k 个簇 $W_1, W_2, \cdots, W_k$，它们的质心依次为 $c_1, c_2, \cdots, c_k$，其中 $c_i = \dfrac{1}{n_i}\sum_{x \in W_i} x$，$n_i$ 是簇 W_i 中数据点的个数。聚类效果的好坏用目标函数 J 表示：

$$J = \sum_{i=1}^{k}\sum_{j=1}^{n_i} d_{ij}(x_j, c_i) \qquad \text{（公式 7-21）}$$

公式 7-21 中，$d_{ij}(x_j, c_i)$ 是 x_j 与 c_i 之间的欧氏距离。目标函数 J 其实就是每个数据点与所在簇的质心的距离之和，所以 J 值越小，簇就越紧凑、越相对独立。因此，算法通过不断优化 J 的取值来寻求好的聚类方案，当 J 取极小值时，对应的聚类方案即为最优方案。

K - means 算法步骤如下：

输入：数据 $X = \{x_1, x_2, \cdots, x_n\}$ 集，聚类数目 k；

输出：k 个类 $C_j, j = 1,2,\cdots,k$ 簇；

步骤 1：令 $I = 1$，随机选取 k 个数据点作为 k 个类簇的初始簇中心，$m_j(I), j = 1,2,\cdots,$

k；

步骤2：计算每一个数据点与这 k 个簇中心的举例 $d(x_i, m_j(I))$，$i = 1,2,\cdots,n$，$j = 1,2,\cdots,k$，如果满足

$$d(x_i, m_j(I))\min\{d(x_i, m_j(I)), J = 1,2,\cdots,k\}$$

则 $x_i \in C_j$；

步骤3：计算 k 各新的聚类中心

$$m_j(I+1) = \frac{1}{N_j}\sum_{\substack{i=1 \\ x_i \in C_j}}^{N_j} x_i, j = 1,2,\cdots,k;$$

步骤4：判断：若 $m_j(I+1) \neq m_j(I)$，$j = 1,2,\cdots,k$，则 $I = I+1$，返回步骤2；否则算法结束；

步骤5：输出聚类结果，算法结束。

以下通过一个例子说明 K－means 算法的工作过程。

给定数据样本集合为 $S = \{2, 7, 25, 21, 13, 27, 30, 20, 16, 9\}$，将 S 分为3类。

第一次迭代：随机选择3个初始中心为 $z_1 = 2, z_2 = 16, z_3 = 30$。计算各个样本点到三个样本中心的距离。因为 $|7-2| < |7-16| < |7-30|$，所以将7划分为 z_1 所属的类 C_1。同理，将9、13、18、22划分为 z_2 所属的类 C_2，将24、27划分为 z_3 所属的类 C_3。

根据划分后的数据对象，重新计算每个类的聚类中心：

$$z_1' = \frac{2+7+9}{3} = 6, z_2' = \frac{9+13+16+18+22}{5} = 15.6, z_3' = \frac{24+27+30}{3} = 27$$

第二次迭代，用新的聚类中心对数据集重新进行划分，划分之后结果为：

$C_1 = \{2,7,9\}$，$C_2 = \{13,16,18\}$，$C_3 = \{22,24,27,30\}$

重新计算聚类中心：

$$z_1' = \frac{2+7+9}{3} = 6, z_2' = \frac{13+16+18}{5} = 15.67, z_3' = \frac{22+24+27+30}{3} = 25.75$$

第三次迭代，根据新的聚类中心对数据集进行划分。划分的结果与第二次迭代的结果一样，迭代过程停止。得到的最优的三个聚类中心为：

$C_1 = \{2,7,9\}$，$C_2 = \{13,16,18\}$，$C_3 = \{22,24,27,30\}$

下面再以一组二维数据为例，简要说明 K－means 算法的聚类过程。

（1）在空间取 $n = 12$ 个数据对象，k 个数据作为初始簇中心点，如图7－2所示。

（2）计算数据对象到各个聚类中心的距离，并将每个对象分配给最近的簇，分配结果如图7－3所示。

（3）更新簇的中心值，如图7－4所示。

（4）根据不断更新簇的中心值后将数据点重新分配，最终聚类结果如图7－5所示。

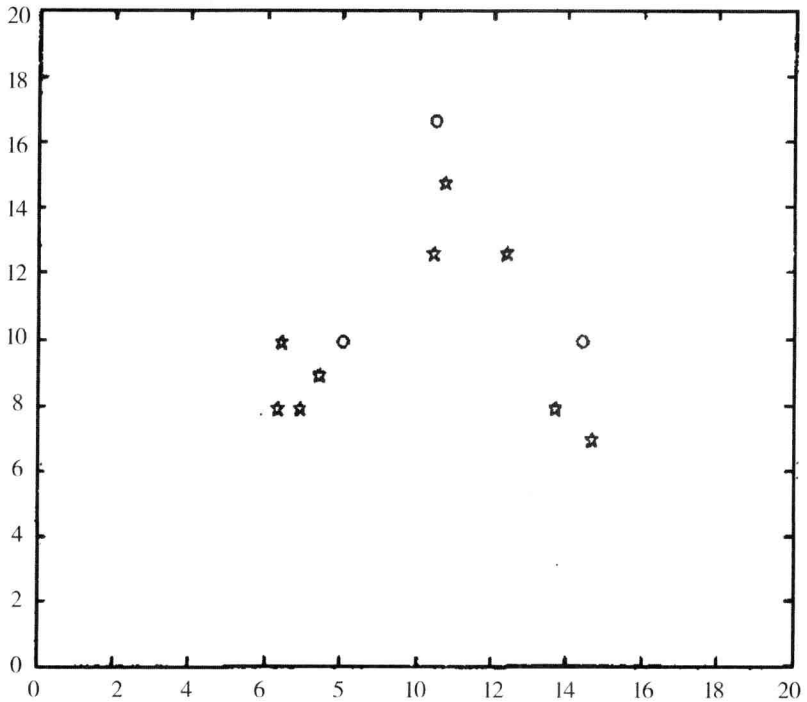


图 7－2　数据分布

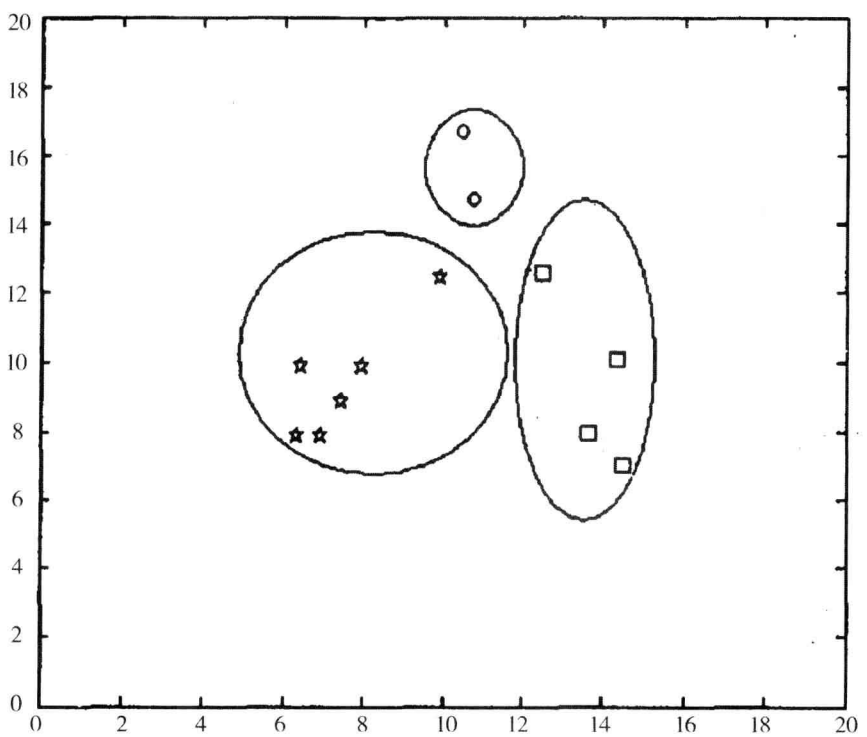


图 7－3　数据初次分配结果

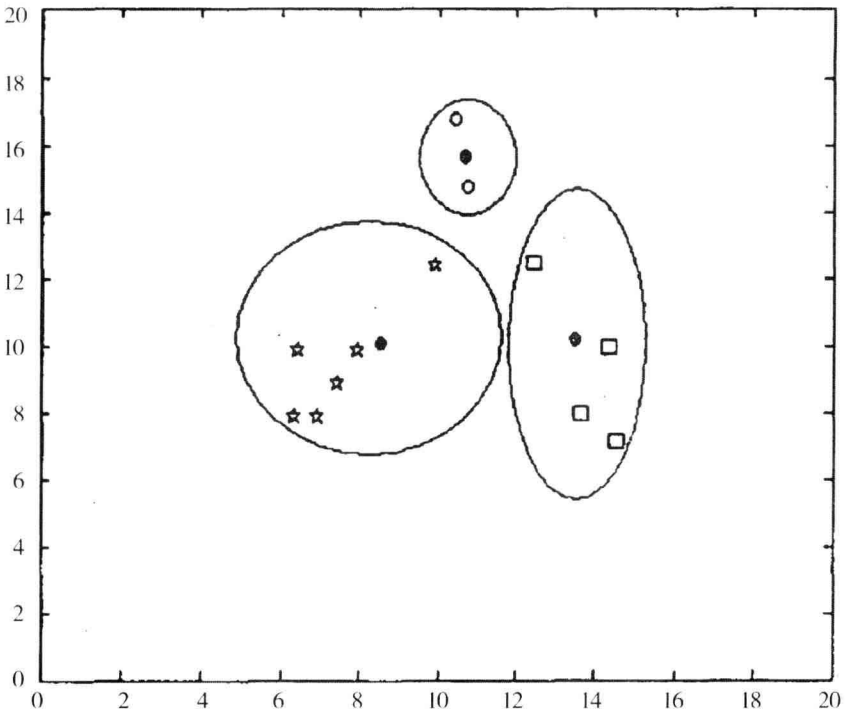


图 7-4　更新数据聚类中心

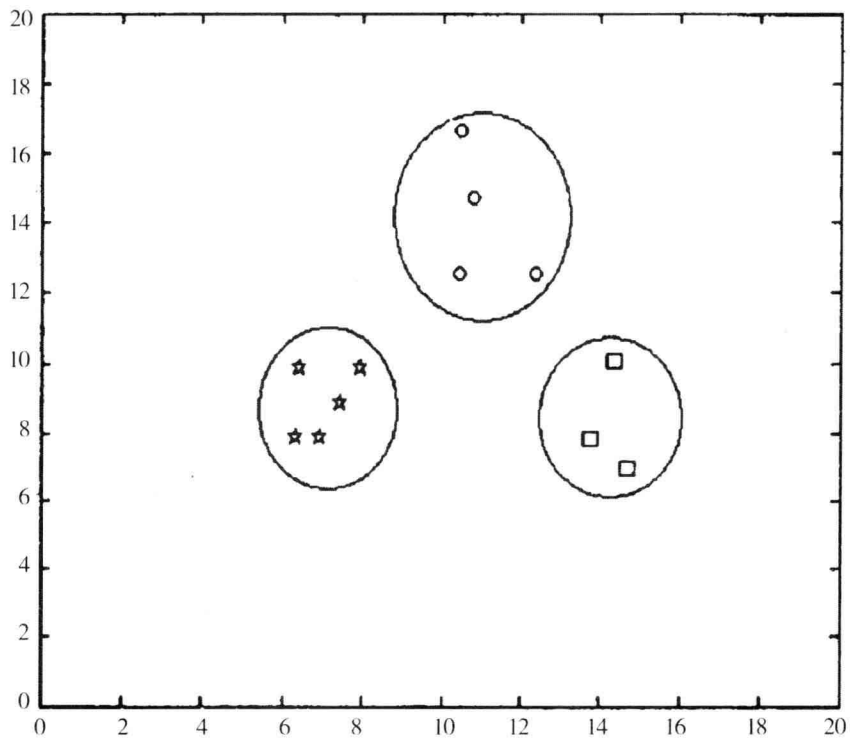


图 7-5　聚类最终结果

7.5.2 K－means 算法优缺点分析

K－means 算法是一种基于划分的聚类算法，它通过不断的迭代过程来进行聚类，当算法收敛到一个结束条件时就终止迭代过程，输出聚类结果。由于其算法思想简便，又容易实现，因此 K－means 算法已成为一种目前最常用的聚类算法之一。然而 K－means 过分依赖于初始中心点的选取，且容易受噪音点的影响。为解决这一问题，出现了各种基于全局最优化思想的 K－means 方法，比如模拟退火算法、遗传算法等。然而这些技术并没有得到广泛认可，在许多实际应用中还是反复利用 K－means 算法来解决问题。

K－means 的优点是：简单易行，不必确定距离矩阵，具有高效性，对于大数据集，算法的可伸缩性较好，具有高效性，因为它的计算复杂度是 $O(nkt)$，其中 n 为对象个数，k 为聚类个数，t 为迭代次数，通常 $t \leqslant n, k \leqslant n$，因此它的复杂度通常也用 $O(n)$ 表示。

K－means 算法采用迭代式的过程对样本点进行分配来寻求最终的聚类结果。其终止条件是所有样本的位置不再变化。其迭代过程可以概括如下：（1）分配样本点，即对每个样本点，将其分配到与其距离最近的簇中心所在的类簇中；（2）重新计算簇中心，对于每一个重新分配后的类簇，重新计算其簇中心。

和大多数的聚类算法一样，K－means 算法也有其自身的局限，主要局限如下：

（1）K－means 算法中的聚类数目即 K 值需要由用户预先给出。从 K－means 算法的算法流程中可以看出，K 值作为一个需要预先确定的参数，在已知的前提下才能执行 K－means 算法，而在实际应用中，需要聚类的数据究竟要分成多少个类别，往往不是被用户所知的。当聚类数目不被人所知的情况下，人们往往需要结合其他算法来获取聚类数目，即 K 值。往往获取 K 值的代价要比 K－means 算法的代价大得多，因此 K 值的不确定性是 K－means 算法的一个很大的不足之处。

（2）K－means 算法严重依赖于初始簇中心点的选取。K－means 算法随机的选取 K 个初始簇中心点，并针对这 K 个簇中心点进行迭代运算，即重新分配数据点和重新计算簇中心的运算，直到所有的数据点位置不再变化或聚类误差准则函数不再变化。这样就导致了 K－means 算法对初始簇中心点的严重依赖性。初始簇中心点选取不当很容易造成聚类结果陷入局部最优解或导致错误的聚类结果。

（3）K－means 算法的聚类结果容易受噪音点数据的影响。在 K－means 算法中，每次对于簇中心的重新计算，都是通过对每一个类簇中所有数据点求均值。这样，当数据集中存在噪音点数据时，均值点的计算将导致聚类中心（即簇中心）偏离数据真正密集的区域，而趋向噪音点数据会导致聚类结果的不准确。因此，当数据集中存在远离所有数据点的噪音点时，聚类结果将在很大程度上受这些噪音点的影响，导致聚类结果的错误，所以 K－means算法对噪声点和孤立点非常敏感。

（4）K－means 算法无法发现任意形状的簇。K－means 算法采用距离函数作为度量数据点间相似度的方法，这里的距离函数多采用欧氏距离，同时采用聚类误差平方和准则函数作为聚类准则函数。对于基于欧式距离的聚类算法而言，其只能发现数据点分布较均匀的类球状簇；对于聚类误差平方和准则函数而言，当类簇大小差别较大，形状较不规则时，容易造成对较大的类簇进行分割来达到目标函数取极小值的目的，因此容易造成错误的聚类结果。

（5）K－means 算法不适用于大数据量的聚类问题。K－means 算法的每次迭代过程都

要调整簇中心及重新分配数据点，因此，当数据量比较大的时候，这些迭代过程的计算量是相当大的，算法的时间开销也是巨大的。由于需要大量的计算时间，K－means 算法在待聚类数据量较大的时候并不适用。

7.6 K－means 算法应用实例

7.6.1 K－means 算法的改进

对于传统的 K－means 算法存在的缺点和不足，研究者从各种不同角度提出了许多改进方法。例如 Bradley 等通过改进算法，降低了聚类结果对于初始聚类质心选取的依赖程度；Pelleg 等科学地提出了新的 K－means 算法，这个算法改进的优点是加速了迭代过程的速度；Dhillon 等使聚类算法性能得到了提高，他是利用聚类过程中重新计算聚类质心的方法；Strehl 则利用图论的思想改进了算法聚类结果的不平衡性。

K－means 算法，对于初始聚类质心的选取有很强的依赖性，初始聚类质心选取的好坏，能够影响到聚类结果是否是有效的。首先需要根据初始聚类质心来确定一个初始划分，然后优化初始划分，所以 K－means 算法的聚类结果对初始质心选取的依赖是很强的。因此，为了更好地解决这一问题，我们可以根据实际生活经验以及数据统计来进行判断，最大可能性地使初始质心点处在事先选取的各个类或簇中，以保证聚类结果的合理性。对此，针对证券业的行业特点，采用将对证券公司贡献度最为关键的指标——手续费收入和资金余额作为未来簇的主要特征，根据证券业行业特点和实际经验，将客户划分为优质客户、潜力客户、一般客户 3 类，即 $k=3$，通过确定合理的初始中心点，优化算法，以便得到较好的聚类效果。其算法改进如下：

步骤：

（1）只对手续费收入、资金余额数据对象进行简单聚类，分为 3 类；

（2）将 3 个类或簇分别算出质心，从数据集中找出与各个质心最为相同或者相近的数据对象作为 3 个类或簇的初始中心点，假设为 a，b，c；

（3）将编号为 a，b，c 的数据对象作为四个变量属性聚类的初始质心；

（4）输入其他数据对象，依次分配给 a，b，c 这 3 个质心最为相似的簇；

（5）更新重新分配的簇的质心；

（6）重复（4）、（5）步；

（7）计算准则函数 $E=\sum_{i=1}^{k}\sum_{a_i\in C_j}|a_i-O_j|^2$；（公式 7－22）

（8）直到 E 收敛或者簇的质心不再发生变化为止。

7.6.2 客户细分模型的建立

客户细分是客户关系管理重要的实现手段之一，采用基于客户交易行为的客户细分是一个可选的方法。针对证券业庞大的客户群，通过一定的划分方法，将客户分为优质客户、潜力客户、一般客户，有利于证券公司提供差异化的服务，有利于客户关系管理的成功实施。

（1）模型变量的解释

客户细分的变量属性有动态属性，也有静态属性。数据挖掘主要提取动态属性，如：手续费收入、息差收入、资金余额和操作频率，而去除与数据挖掘相关性不强的静态属性，如证券名称、交易时间、客户姓名、客户年龄等，如表7-1所示。

表7-1　变量属性表

变量名称	属性代码	变量类型	变量长度
证券名称	securities name	text	8
客户姓名	customer name	text	8
交易时间	Trading hours	time	8
手续费收入	fee income	value	4
息差收入	margins income	value	not null
资金余额	found balance	value	not null
操作频率	operating frequency	value	not null

（2）数据的预处理

数据挖掘中尤为重要的一个关键环节是数据预处理，为了挖掘出更为有效的客户信息就必须提供准确、干净、标准、整洁的数据。但是在现实生活中，数据对象并不都是规范的，大多会有噪音、冗余、数据不一致等问题出现。这些问题的出现可能是人为的遗漏，也可能是误删或者是机器设备导致的故障等等。为了能够提高数据挖掘的质量、效率，就必须对这些存在缺陷的、不完整的数据进行预处理，使得数据挖掘过程更简单、更有效、更实用，结果更能准确地反映客观情况，因为数据中的这些缺陷很有可能导致数据挖掘最终产生不良的后果。数据预处理可以概括为以下三个阶段：

①数据选取

由于从数据库中随机选取的历史交易数据属性繁多、数量巨大，这就需要人为地设定标准以便能从庞大的数据库中便携地选取合适的数据。

②数据清理

数据清理，这个过程主要是为了帮助除去选取数据中的噪音，并合理解决数据丢失的问题。噪音处理、空值处理以及不一致数据处理等是数据清理的重点。这三种数据处理都可导致挖掘结果的准确度降低。

本节采用K-means算法对客户的历史交易数据进行聚类分析，在使用K-means算法对交易数据的变量属性进行聚类分析前，对大量交易数据进行清洗是必不可少的环节。这是因为客户交易数据大多是数值型数据并且带有大量的无价值的冗余信息。

③数据变换

数据的观侧值可能相差很大，因为数据库中的各个数据对象具有各种不同的性质属性。通常都有不同的度量单位。所以，绝对值相对较少的变量起不到应有的作用，绝对值较大的变量可能会影响到绝对值较小的变量。为了保证各变量在聚类分析中所处的地位相同，有必要对数据对象进行数据变换。中心化变换与标准化变换就是两种常见的变换方法。

中心化变换的目的是使各种变量的观侧值都具备相同的基点，公式是：

$$x_{ij}' = x_{ij} \cdot x_j \qquad \text{（公式 7-23）}$$

其中，x_{ij} 是第 i 个数据对象所包含的第 j 个属性：$x_j = \frac{1}{n}\sum x_{ij}; 1 \leqslant i,j \leqslant n$。

标准化变换的目的是在上例中心化变换的基础上再作数据变换，使各种变量的属性变换范围相等。以下是标准差的标准化变换：

$$x_{ij}^n = \frac{X'_{ij}}{S_j} \qquad \text{（公式 7-24）}$$

其中，

$$S_j = \left[\frac{1}{n-1}\sum (x_{ij} - x_j)^2\right]^2 \qquad \text{（公式 7-25）}$$

在经过中心化、标准化变换后，各数据对象的变量属性的均值为 0，且标准差均为 1。也就是说各变量属性的基点相同，并且变化范围也相同。

（3）改进算法的数据示例

本例所选的具体数据选自济南某证券公司柜面系统中 2008 年部分客户历史交易数，经过数据的选取、清理后选取了 18 个客户信息，包含手续费收入、资金余额、息收入、操作频率等变量，进行改进算法的验证，如表 7-2 所示。

表 7-2　客户信息表

资金序号	手续费收入	资金金额	息差收入	操作频率
0001	0.00	1038.24	2.43	0
0002	13.30	12.43	0.03	1
0003	8806.3	7066.88	0.25	603
0004	214.4	27.48	0.07	17
0005	4174.44	55.50	0.14	274
0006	435.44	1.24	0.00	20
0007	0.00	462.44	1.14	0
0008	10.82	12.73	0.03	1
0009	87.54	6.25	0.02	10
0010	170 003.45	221 489.10	3.06	402
0011	329.51	7539.36	0.45	8
0012	999.12	35 265.22	4.29	9
0013	0.00	11.42	0.03	0
0014	713.41	16 924.23	2.61	28
0015	0.00	780.85	1.92	0
0016	1159.24	41 110.71	0.73	26
0017	5241.03	24 586.32	50.26	267
0018	128.66	289.36	0.56	38

先仅对手续费收入、资金余额两个变量进行聚类，找出第二次聚类的初始质心点，用 SPSS16.0 软件进行聚类分析，具体聚类结果如表 7-3～表 7-6 所示。

表 7 - 3　Z 变换数据结构

c_ name	f_ income	f_ balance
0001	- 0. 1966	- 0. 3613
0002	- 0. 4936	- 0. 3810
0003	1. 5055	- 0. 2453
0004	- 0. 4479	- 0. 3807
0005	0. 4525	- 0. 3802
0006	- 0. 3976	- 0. 3812
0007	- 0. 4966	- 0. 3724
0008	- 0. 4941	- 0. 3810
0009	- 0. 4767	- 0. 3811
0010	3. 3691	3. 8802
0011	- 0. 4217	- 0. 2362
0012	- 0. 2694	0. 2973
0013	- 0. 4966	- 0. 3810
0014	- 0. 3344	- 0. 0556
0015	- 0. 4966	- 0. 3662
0016	- 0. 2330	0. 4097
0017	0. 6950	0. 0918
0018	- 0. 4673	- 0. 3757

表 7 - 4　初始聚类中心

	Cluster		
	1	2	3
Zscóre (f_ income)	- 0. 49658	3. 36910	1. 50550
Zscore (f_ balance)	- 0. 38103	3. 88024	- 0. 24529

表 7 - 5　最终聚类中心

	Cluster		
	1	2	3
Zscore (f_ income)	- 0. 43014	3. 36910	0. 88431
Zscore (f_ balance)	- 0. 23904	3. 88024	- 0. 17789

表 7-6　每类样品数目表

Cluster	1	14.000
	2	1.000
	3	3.000
Valid		18.000
Missing		0.000

从表 7-3～表 7-6 中，可以得出聚类结果：

最后聚成 3 簇，每一簇的数据个数和簇的质心分别为：cluster1 14 个数据变量，质心分别为（-0.43014，-0.23904）；cluster2 1 个数据变量，质心分别为（3.36910，3.88024）；cluster3 3 个数据变量，质心分别为（0.88431，-0.17789）。

根据相同或者相似的原则选定簇的三个初始质心分别为：0011（329.51，7539.36）、0010（17003.45，221489.10）、0003（8806.30，7066.88）。

则初始聚类质心的选取分别为 0011、0010、0003。

再对包含四个变量属性的数据对象进行聚类，用 SPSS16.0 软件聚类，其结果如表 7-7～表 7-10 所示。

表 7-7　Z 变换数据结构

c_ name	f_ income	f_ balance	m_ income	o_ frenquency
0001	-0.4966	-0.3613	-0.1156	-0.5444
0002	-0.4936	-0.3810	-0.3212	-0.5686
0003	1.5055	-0.2453	-0.3024	2.9231
0004	-0.4479	-0.3807	-0.3178	-0.4466
0005	0.4525	-0.3802	-0.3118	1.0312
0006	-0.3976	-0.3812	-0.3238	-0.4294
0007	-0.4966	-0.3724	-0.2261	-0.5444
0008	-0.4941	-0.3810	-0.3212	-0.5386
0009	-0.4767	-0.3811	-0.3221	-0.4869
0010	3.3691	3.8802	-0.0616	1.7673
0011	-0.4217	-0.2362	-0.2852	-0.4984
0012	-0.2694	0.2973	0.0438	-0.4926
0013	-0.4966	-0.3810	-0.3212	-0.5444
0014	-0.3344	-0.0556	-0.1002	-0.3834
0015	-0.4966	-0.3662	-0.1593	-0.5444
0016	-0.2330	0.4097	-0.2612	-0.3949
0017	0.6950	0.0918	3.9826	0.9910
0018	-0.4673	-0.3757	-0.2758	-0.3259

表 7 - 8　初始聚类中心

	Cluster		
	1	2	3
Zscore（f_ income）	-0.42167	3.36910	1.50550
Zscore（f_ balance）	-0.23620	3.88024	-0.24529
Zscore（m_ income）	-0.28523	-0.06160	-0.30236
Zscore（o_ frenquency）	-0.49837	1.76729	2.92312

表 7 - 9　最终聚类中心

Cluster			
	1	2	3
Zscore（f_ income）	0.69495	3.36910	-0.25400
Zscore（f_ balance）	0.09179	3.88024	-0.24825
Zscore（m_ income）	3.98259	-0.06160	-0.24506
Zscore（o_ frenquency）	0.99099	1.76729	-0.17239

表 7 - 10　每类样品数目表

Cluster	1	3.000
	2	1.000
	3	15.000
Valid		18.000
Missing		0.000

由表 7 - 7 ~ 表 7 - 10 可知，算法最终形成了 3 个细分群分别为：

Cluster1 包括 0003，0017，0005 三个数据对象；

Cluster2 包括 0010 一个数据对象；

Cluster3 则包括其他 14 个数据对象，且所有的数据对象都是有效的，没有缺失。

因此，改进后的 K - means 算法，明确确定了初始质心点的选取方法。根据证券业企业利润的贡献度，采取对手续费收入和资金余额这两个数据对象进行预聚类，将其聚类结果作为下一步聚类的初始质心的方法。与原算法相比，加快了 K - means 算法聚类的准确性与合理性，很好地解决了孤立点的问题，从最终的聚类结果看，两次聚类的结果差异不大，表明从变量属性贡献度来聚类客户是行之有效的。改进后的算法解决了以往聚类初始中心点如何选取的难题，将原先的无指导式的学习变成有目的地学习，最终结果有利于实现聚类分析的客观性、稳定性，最终达到合理的客户细分。

第 8 章　数据挖掘相关技术

8.1　粗糙集技术

8.1.1　概述

在现实生活中，我们所接触到的和所获取的信息，其不确定性是绝对的，其确定性是相对的。如果我们所获得的信息是确定的，则我们会根据这些信息来做出各种各样的决策。如果我们要决策的事物能够通过以前确定的信息准确做出判断，那么这种决策非常容易。如果要决策的事物不能通过已知信息做出准确的判断，则决策里面就含有不确定的因素。在自然科学、社会科学和工程技术的很多领域中，都不同程度地涉及对不确定因素的处理。从实际系统中采集出来的样本往往带有一些干扰，通常我们没有考虑这些干扰，利用已经设计的精确数学模型来对其进行求解和仿真，然后再把所得到的参数用来控制实际系统。结果表明这种处理方法不理想。因为在建立数学模型求解时，利用假设消除或回避了这种不确定性。反之，如果正视它，对这些信息进行合适的处理，得到的结果则会比较理想。因此在生活和生产中处处存在对不确定问题的处理。

在较长时间里，精确数学及随机数学描述自然界多种事物的运动规律，但是，在客观世界中还普遍存在着大量的模糊现象。以前人们回避它，但是，由于现代科技所面对的系统日益复杂，模糊性总是伴随着复杂性出现。模糊集注重描述信息的含糊程度，即属于同一类别的不同对象的隶属关系。模糊集是不可计算的，它没有给出数学公式描述这一含糊概念，故无法计算出它的边界线上的具体的含糊元素数目，只能凭借人们的经验来给出。同概率论一样，模糊集在处理不确定信息时必须需要先验知识，也就是需要模糊隶属函数。这一缺点也限制了模糊理论的发展。

20 世纪 70 年代初，波兰学者 Pawlakz 领导的波兰科学院和华沙大学的研究小组，开始对信息系统逻辑特性进行长期基础性的研究。1982 年粗糙集理论正式提出。该理论是一种刻画不完整性和不确定性的数学工具，能够有效地分析和处理不精确、不一致、不完整等各种不完备信息，并从中发现隐含的知识，揭示潜在的规律。粗糙集理论不需要先验知识，而且它也是一个强大的数据分析工具，所以粗糙集理论成为目前解决不确定问题的重要工具。

8.1.2 知识表达系统

下面给出粗糙集的基本定义。

定义 8.1 设一个信息表知识表达系统 $S=(U,R,V,f)$，其中：

U 表示数据对象的非空有限集合，称为论域；

A 表示属性的非空有限集合，$A=C\cup D, C\cap D=\phi$，C 称为条件属性，D 称为决策属性；

V 表示属性值的集，且 V_i 是属性 A_i 的值域；

f 表示信息函数，又称 CD 决策表。

定义 8.2 若非空属性子集 $C\subseteq A$ 时，称 C 的不可分辨关系（不可区分关系）$IND(C)$ 是 U 上的等价关系（满足自反性、对称性和传递性）。其中：

$IND(C)=\{(x,y)\in U^2 \mid \forall a\in C, f(x,a)=f(y,a)\}$。

为方便起见，$IND(C)$ 可以用 C 代替，所有等价关系类的集合记为 U/C，称为 C 到 U 上的划分，划分的等价类称为基本集。

定义 8.3 令 $X\subseteq U$，R 为 U 上的一个等价关系，如果 X 能用 R 的范畴确切的描述，称 X 是 R 可定义的；否则称 X 是 R 不可定义的。R 可定义集称为精确集，R 不可定义集称为 R 非精确集或 R 粗糙集。

对于给定论域空间中的样例子集以及不可分辨关系，为了描述其他的样例是否属于该样例子集做出明确的判断，粗糙集理论给出了上近似集、下近似集、正域和负域、边界域的概念。

定义 8.4 设 $X\subseteq U$，R 是一个定义在 U 上的等价关系。如果：

① $R_*(X)=U\{Y\in U/R: Y\subseteq X\}$，称 $R_*(X)$ 为 X 的 R 下近似集；

② $R^*(X)=U\{Y\in U/R: Y\cap X\neq\phi\}$，称 $R^*(X)$ 为 X 的 R 上近似集；

③ $POS_R(X)=R_*(X)$，称 $POS_R(X)$ 为 X 的正域；

$NEG_R(X)=U/R_*(X)$，称 $NEG_R(X)$ 为 X 的 R 负域；

④ $BN_R(X)=R^*(X)/R_*(X)$，称 $BN_R(X)$ 为集合 X 的 R 边界域。

由此定义，我们可以看出，下近似值集是根据属性子集 R，中所有一定能归入集合 X 的元组的集合，上近似值是根据属性子集 R，U 中所有一定和可能归入集合 X 的元组的集合；正域是所有一定能归入集合 X 的元组集合；负域是所有不能确定一定归入集合 X 的元组的集合；边界域是不确定域，也就是不能肯定归入集合 X，也不能肯定地归入 X 的补集。而且，我们规定，当且仅当 $R^*(X)=R_*(X)$，称集合 X 是 R 的可定义集；反之当且仅当 $R^*(X)\neq R_*(X)$ 时，即边界域不为空时，称集合 X 是 R 的粗糙集。

8.1.3 粗糙集的主要研究内容

①可变精度粗糖集的研究

Pawlak 教授提出的传统粗糖集理论模型虽然是一种比较有效地软计算工具，但是也存在一定的局限性，有效地扩展这些概念是十分必要的，所以很多学者推广了这一模型，其中提出的比较经典的是相容模型和可变精度粗糖集模型。由于经典的粗糖集模型建立的基础是强不可分辨关系，也就是处理的样本分类必须是完全确切或肯定的，严格按照等价类

进行分类，亦是“包含”和“不包含”关系，但是实际处理的数据存在很多不确定因素，尤其是在文本分类当中，所以有必要考虑相容粗糖集模型。相容模型特别考虑了相容关系或相似关系。相容关系主要研究的是在属性集中的条件下定义相容关系，相容关系满足自反性和对称性，属性集中满足这个相容关系的对象的集合称为相容类，显然，相容类对数据集的划分是相互重叠的，构成了对原数据集的覆盖。数据库当中的数据对象很多时候的属性值是缺失的，不可分辨关系无法应对这种情况。M. kryszkiewicz 等学者提出了一种相似关系来代替，但这样相似关系虽然满足自反性和对称性，但是不具有传递性。可变精度模型（VPRS）根据经典粗糖集模型，在此基础上引入了一个阈值（$0 \leqslant r \leqslant 0.5$），表示允许一定程度的错误分类率，也就是当所给等价类在一定的程度上包含于集合 X 时，就认为这个对象属于集合 X。这一推广得到了广泛的应用，因为在现实处理数据的时候，很多情况下数据是不确定的，基于此很多学者对可变精度模型进行了研究，并取得了一定的成绩，其中最著名的就是 Ziarko 提出的一种可变精度模型（VPRS）。

②连续属性的离散化处理

在现实的决策表中经常存在连续属性的问题，而粗糖集理论的基础是集合论，却难以直接处理这些连续的属性。因此连续属性的离散化处理成为粗糖集理论的研究热点之一，其研究的出发点是在保持决策结果不发生变化的情况下，对决策表中的连续属性进行简化处理，得到简化的决策表。近年来，国内外学者对基于粗糖集理论的离散化方法进行了深入的研究，比较有代表性的算法有粗糖集工具软件 Rosetta 提供的 Naive Scaler 算法和改进的 Semi Naive Scaler 算法。这两种算法直接根据决策表本身进行离散，不需要额外参数，但每次仅考虑单个属性，所以这两种方法属于局部离散化范畴。最直观且容易被理解的整体离散化方法是 H. S. Nguyen 等提出的布尔逻辑和粗糖集理论相结合的离散化方法，但这种算法的时间和空间复杂度都相对较高。还有一些方法是基于在离散的过程中是否使用类别属性做参考，分为有监督离散化和无监督离散化。例如，等频区间法和等宽区间法就是典型的无监督离散化方法，它们都属于盲目算法，即不考虑属性类别，反之，那些应用类别信息进行离散化处理的方法则称为有监督离散化方法，典型的有监督离散化方法有信息熵法、聚类法和决策树法等。目前连续属性的离散化存在的不足是缺乏递增的方法，解决这个问题是离散化方法研究的方向之一。

③粗糙集理论算法的研究

粗糙集理论的算法主要集中于规则提取、属性约简和数据挖掘等方面，粗糙集理论在进行规则提取和数据挖掘的过程中最关键的步骤就是决策表的属性约简，所以属性约简算法的性能直接决定了利用粗糖集理论对数据进行分类能力的重要指标。目前主要的属性约简算法有基于差别矩阵的属性约简算法、基于属性重要性的约简算法、基于特征选择的属性约简算法等，为了增加分类的稳健性，提出了动态约简的方法。然而，S. K. MWong 和 W. Ziarko 已经证明属性约简算法和连续属性离散化方法一样都属于 NP - hard 问题，主要是由于属性的组合爆炸问题引起的。所以一般我们只需找到一个最优或近似最优约简即可。

8.1.4 属性约简

属性约简是粗糙集理论的核心内容之一，在数据挖掘领域具有重要的应用意义。所谓约简就是在保持属性集分类能力不变的条件下，删除其中不相关的或不重要的属性。在定

义约简之前，我们先做如下定义：

设 $T=(U,A,V,f)$ 是一个信息系统，$a\in A$ 是一个属性，如果属性集 $A-\{a\}$ 与属性集 A 的区分能力相同，则称属性 a 是不必要的，否则称 a 是必要的。如果任意 $a\in A$ 都是必要的，则称 A 是独立的，否则称 A 是依赖的。

如果属性子集 $B\subseteq A$，如果 B 是独立的并且 B 与 A 的区分能力相同，则 B 是 A 的一个约简，记为 $RED(A)$。显然，A 可以有多个约简。A 中所有必要的属性的集合称为 A 的核，记为 $CORE(A)$。

定理 8.1　设 $T=(U,A,V,f)$ 是一个信息系统，则属性集 A 的所有约简的交集是 A 的属性核，即

$$CORE(A)=\cap RED(A)$$

从定理 8.1 可以看出，核这个概念的用处有两个方面：第一，属性核包含在所有属性约简中，计算约简时可以将核作为约简计算的起点；第二，在属性约简的过程中，属性核作为属性集中最核心的部分，不能被删除。

相对约简和相对核的概念反映了一个分类与另一个分类的关系，这两个概念是决策表属性约简的基础。

定义 8.5　设 $T=(U,C\cup D,V,f)$ 是一个决策表，D 的 C 正域记作 $POS_C(D)$，定义为

$$POS_C(D)=\underset{X\in U/P}{U}CX$$

定义 8.6　设 $T=(U,C\cup D,V,f)$ 是一个决策表，$a\in C$ 是一个条件属性，若

$$POS_C(D)=POS_{C-\{a\}}(D)$$

则称 a 为 C 中 D 必要，否则称为 C 中 D 不必要。

如果任意 $a\in C$ 都是 D 必要的，则称 C 是独立的，否则称 C 是依赖的。

定义 8.7　设 $T=(U,C\cup D,V,f)$，若 $B\subseteq C$ 是独立的，$a\in C$，且 $POS_C(D)=POS_{C-\{a\}}(D)$ 时，称 B 是 C 关于 D 的相对约简。

C 中所有相对 D 不能去掉的属性构成的集合称为 C 的 D 核，记作 $CORE_D(C)$。所有的 C 的 D 相对约简的集合记作 $RED_D(C)$。

定理 8.2　设 $T=(U,C\cup D,V,f)$ 是一个决策表，若 $B\subseteq C$ 是独立的，则属性集 C 的所有相对约简的交集是 C 的属性核，即

$$CORE_D(C)=\cap RED_DC$$

一般属性约简是在不改变信息系统中属性集对象的分类能力的前提下去掉不相关的属性，而相对约简是在不改变将决策表中的对象划分到另一个独立的划分中的区分能力的前提下消去冗余的条件属性。

8.2　模糊集技术

在自然科学或社会科学研究中，存在着许多定义不很严格或者说具有模糊性的概念。所谓模糊性是指客观事物处于共维条件下的差异在中介过渡时所呈现的“亦此亦彼”性。因为客观事物之间的差异并不一定具有中介过渡性。当事物处于共维条件下，它们的差异才呈现出中介过渡。相互差异着的事物必须具有某种本质上共同的东西作为维，或简称为共维，才能呈现中介过渡性。中介过渡的两端称为两极，具有相反的性质、作用和倾向。

在同一维的两极之间，有许多中介点、中介状态，通过中介相互过渡与转化，在两极之间的中介过渡往往具有连续性。

模糊理论是在美国加州大学伯克利分校电气工程系的 L. A. Zadeh 教授于 1965 年创立的模糊集合理论的数学基础上发展起来的，主要包括模糊集合理论、模糊逻辑、模糊推理和模糊控制等方面的内容。

模糊集理论中的“模糊”主要指的是那些外延不明确的概念。例如“青年人”，你能在年龄轴上画两道线，表明在两道线内的就是青年人，在其外的就截然不是青年人吗？显然不能，人的生命是一个连续的过程，一个人从少年走向青年是日复一日积累的，在经济科学和管理科学中，这样的概念也处处可见。比如“通货膨胀”，如果说物价上涨率超过 10% 就意味着通货膨胀，那么 9.9% 的情形呢？就绝对不是吗？而且物价上涨 20% 的情形和物价上涨 200% 的情形又如何在外延上加以区别呢？我们生活中有许许多多类似这样外延不分明的概念。模糊数和传统概率理论中的随机变量的不同之处就在于，随机变量是反映客观的现象，而模糊数的衡量包含了人的主观的因素。因此，模糊性反映了事件的不确定性，但这种不确定性不同于随机性。随机性反映的是客观上的自然的不确定性，或事件发生的偶然性，而模糊性则反映人们主观理解上的不确定性，即人们对有关事件定义或概念描述在语言意义理解上的不确定性。

8.2.1 模糊集合与隶属度函数

先考虑普通集合的情况，设 U 表示一些对象的集合，称之为论域。对于 U 的一个子集 A，可以用它的特征函数来表示。令

$$\chi_A(u) = \begin{cases} 1, u \in A \\ 0, u \notin A \end{cases}$$

χ_A 是定义于 U 上取之于 $\{0, 1\}$ 的函数，称为集合 A 的特征函数。对于 $u \in U$，若 $\chi_A(u) = 1$，则说 u 不是 A 中的元素。由此出发，给出模糊集合的定义。

定义 8.8　设 U 是论域，U 上的一个模糊集合 A 由 U 上的一个实值函数 $\mu_A U \to [0,1]$ 表示。对于 $u \in U$，$\mu_A(u)$ 称为 u 对于 A 的隶属度，而 μ_A 称为 A 的隶属度函数。为了方便起见，用 $A(u)$ 来代替 $\mu_A(u)$。

由上述定义可知，模糊集合 A 是由其隶属函数 $\mu_A(u)$ 定义的，对于论域的一个元素 u 和 U 上的一个模糊子集 A，u 属于 A 的隶属度 $\mu_A(u)$ 即为隶属函数 μ_A 在元素 u 处的值，我们也不再简单地问 u 是“绝对”属于还是不属于 A，而是问 u 在多大程度上隶属于 A；若 $A(u) = 1$，则认为 u 完全属于 A；若 $0 < A(u) < 1$，则说 u 在 $A(u)$ 程度上属于 A，这是在完全属于 A 和完全不属于 A 的元素之间，呈现一种中间过渡状态，或叫连续变化状态。这正是所谓的模糊性。更多的时候要考虑多个模糊集之间的关系，下面定义模糊集的关系与运算。

定义 8.9　设 A 与 B 是 U 上的模糊集。

(1) A 与 B 的并记为 $A \cup B$，其隶属函数为 $(A \cup B)(u) = A(u) \vee B(u)$，其中“$\vee$”表示两者比较后取大值。

(2) A 与 B 的交记为 $A \cap B$，其隶属函数为 $(A \cap B)(u) = A(u) \wedge B(u)$，其中“$\wedge$”表示两者比较后取小值。

(3) A 的余模糊集记为 A^C，其隶属函数为 $A^C(u) = 1 - A(u)$。

(4) 如果 $A(u) \leqslant B(u), \forall u \in U$，则说 A 被 B 包含，记为 $A \subseteq B$。

如果把普通集合看做是模糊集合的特例，那么，立即可以看出普通集合按定义 8.9 的关系及运算与集合论中相应的关系及运算是完全一致的。因此，模糊集的上述关系与运算是对普通集合的关系及运算的推广。

通过上述的定义已经可以认识到所谓模糊集是由它的隶属函数具体表示的。模糊集所表示的外延是不分明的，或者说模糊集所含的元素是模糊的。然而，在处理实际问题的某个时刻，要对模糊概念有个明确的认识与判定，要判断某个元素对模糊集的明确的归属。这就要求模糊集与普通集合可以依某种法则相互转化，模糊集的截集是解决这个问题的一种比较满意的手段。

定义 8.10　设 $A \in F(u)$，任取 $\lambda \in [0,1]$，记 $A_\lambda = \{u \in U \mid A(u) \geqslant \lambda\}$，称 A_λ 为 A 的 λ 截集，其中 λ 称为阀值或置信水平。又记 $A_\lambda^S = \{u \in U \mid A(u) > \lambda\}$，则 A_λ^S 为 A 的 λ 强截集。其中 $F(u)$ 为 U 上的模糊集合的全体。

这样，A_λ 是 U 的一个普通集合，它由那些对模糊集 A 的隶属度不小于水平 λ 的成员所构成。而 A_λ^S 则是由那些对模糊集 A 的隶属度严格大于水平 λ 的成员所构成的。

8.2.2　隶属度函数的确定方法

确定元素对某个模糊集合的隶属度常用的方法有：边界法、模糊统计法、参照法（模糊分布法）、二元对比法、三分法。分别说明如下：

(1) 边界法：虽然模糊集是没有明确边界的，并且论域中元素的隶属度一般也是渐变的，但是客观事物有质变，人们对客观事物的主观反映相应也有质变，这种质变就是隶属度边界取值 0 或 1。

(2) 模糊统计法：可以分为直接统计法和隶属频率统计法。直接统计法就是直接对一群人进行调查，每个人对模糊集中的每个元素进行综合打分，若此元素完全属于该模糊集，则为 100 分。每个人打分后取其平均分（有时还去掉一个最高分，去掉一个最低分后再平均），这个平均分就是隶属度。隶属频率统计法是用类似于确定随机事件的概率的方法来确定隶属度。我们可以对模糊事件作统计试验，先确定一个论域（如 0～100 岁），然后对论域中的模糊集作清晰化的范围估计。对于论域中的具体点而言，它可以在某个范围估计中，也可以不在其中。每一次范围估计都可以看成是一次统计试验，于是便可以计算 x 隶属于模糊集 A 的概率：

$$x \text{ 隶属于模糊集 } A \text{ 的频率} = x \in A \text{ 的次数} / n$$

随着 n 的增大，隶属频率会呈现稳定性，就可以把隶属频率的稳定值作为 x 对模糊集 A 的隶属度。

(3) 参照法：主要是利用一些函数，通过参照比较，选择最能代表所论模糊集的函数作为隶属函数，这些函数也称为模糊分布。常用的一些函数有如下一些类型：偏小型、偏大型、中间型。

(4) 二元对比法：采用对比的方法来确定隶属度的值。例如，对于在“人”论域中考虑“聪明”模糊集合，若 x_1 较 x_2 聪明，则规定 $A(x_1) > A(x_2)$。这种方法还根据具体问题分为相对比较法、对比平均法、择优比较法和优先关系定序法。这种方法是模糊综合评判时常用的一种方法。

(5) 三分法：类似于模糊统计法，也是用随机区间的思想来处理模糊性的试验模型。

需要说明的是：在实际应用中，可以根据具体的问题选择不同的方法，也可以多种方法同时使用，还可以自行设计隶属函数。

8.2.3 模糊关系与模糊推理

（1）模糊关系

描述元素之间是否相关的数学模型称为关系，描述元素之间相关程度的数学模型称为模糊关系。当论域有限时，可用模糊矩阵表示模糊关系。模糊矩阵为模糊关系的运算带来了极大的方便，称为模糊关系的主要运算工具。

定义 8.11　对于给定的两个论域 X 和 Y，有直积空间

$$X \times Y = \{(x,y) \mid x \in X, y \in Y\}$$

式中的模糊子集 R 称为 $X \times Y$ 上的模糊二元关系。对于 $x \in X, y \in Y$，模糊关系 R 的隶属函数由以下映射来确定：

$$\mu_R : X \times Y \to [0,1], (x,y) \mid \mu_R(x,y) \to [0,1]$$

隶属度 $\mu_R(x,y)$ 刻画了 x 和 y 之间的 R 相关程度，记为 $R(x,y)$。

集合 X 到 Y 的模糊关系，称为集合 X 上的模糊关系。通常用模糊矩阵来表示模糊二元关系。当 $X = \{x_i \mid i = 1,2,\cdots,m\}$，$X = \{y_i \mid i = 1,2,\cdots,n\}$ 是有限集合时，则 $X \times Y$ 的模糊关系 R 可用下列 $m \times n$ 的矩阵表示：

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{ij} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2j} & \cdots & r_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ r_{i1} & r_{i2} & \cdots & r_{ij} & \cdots & r_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ r_{m1} & r_{m2} & \cdots & r_{mj} & \cdots & r_{mn} \end{bmatrix}$$

式中 $r_{ij} = \mu_R(x_i, y_j)$。由此表示模糊关系的矩阵，被称为模糊矩阵。由于 μ_R 的取值区间为［0，1］，因此模糊矩阵元素 r_{ij} 的值也都在［0，1］区间。

（2）模糊推理

最简单的 If…then 规则的形式："如果 x 是 A，则 y 是 B。" 复合型 If…then 规则的形式很多，例如：

"If m 是 A 且 x 是 B then y 是 C，否则 z 是 D"；

"If m 是 A 且 x 是 B 且 y 是 C then z 是 D"；

"If m 是 A 或 x 是 B then y 是 C ，或 z 是 D"；

"If m 是 A 且 x 是 B then y 是 C，且 z 是 D"；

"If m 是 A 且 x 是 B then y 是 C，且 z 是 D"。

这里 A，B，C，D 分别是论域 M，X，Y，Z 中模糊集的语义值，If 部分是前提或前件 then 部分是结论或后件。解释 If…then 规则包括以下三个过程：

（1）输入模糊化

确定出 If…then 规则前提中每个命题或断言为真的程度（即隶属度）。

（2）应用模糊算子

如果规则的前提有几部分，则利用模糊算子可以确定出整个前提为真的程度（即整个

前提的隶属度）。

（3）应用于蕴含算子

由前提的隶属度和蕴涵算子，可以确定出结论为真的程度（即结论的隶属度）。If…then 规则还可以推广成其他模式，如：

①If A then B，else C

②If A and B then C，else D

③If A and B or C then D，else F

……

推理就是根据已知的一些命题，按照一定的法则，去推断一个新的命题的思维过程和思维方式。简言之，从已知条件求未知结果的思维过程就是推理。模糊逻辑推理是一种不确定性的推理方法，其基础是模糊逻辑。他是一种以模糊判断为前提，运用模糊语言规则推出一个新的近似的模糊判断结论的方法。

8.3 数据仓库

8.3.1 概述

数据仓库的概念首先是由 W. H. Inmon 在《Building the Data Warehouse》中提出的。文中数据仓库定义为“面向主题的、集成的、非易失的、随时间变化的，用来支持管理人员决策的数据集合”。从数据存储角度，数据仓库是存储数据的一种组织形式；从方法论角度，数据仓库是解决问题的一个过程；从实体的角度，数据仓库由可操作的数据源、数据仓库、数据仓库管理系统、数据仓库分析工具、用户界面组成；从功能角度，数据仓库至少包括三个基本功能：数据获取、数据存储和管理、信息访问。数据仓库工具主要是数据分析工具，有数据挖掘和联机分析处理。两者相辅相成，为数据仓库服务。总之，数据仓库从广义上讲就是一个专门的数据存储介质，用来保存从多个数据库和其他信息源选取的数据，并为上层应用提供统一的用户接口，用来完成对选择数据的检索和分析。根据数据仓库概念的含义，数据仓库拥有以下 4 个特点：

（1）时变的。操作型数据库主要关心当前某一个时间段内的数据，而数据仓库中的数据通常包含历史信息，系统记录了企业从过去某一时点到目前的各个阶段的信息。通过这些信息，可以对企业的发展历程和未来趋势做出定量分析和预测。这一特征具体表现在以下三个方面：

①数据仓库随时间变化不断增加新的数据内容；

②数据仓库随时间变化不断删除无用的数据。这里需要说明的是数据仓库的数据存储时限要远远长于操作环境中数据的时限，以适应决策分析的要求；

③数据仓库中包含的大量综合数据很多是跟时间有密切关系的。例如数据经常按照时间段进行综合或者每隔一定的时间段进行抽样。

（2）相对稳定的。操作型数据库中的数据通常实时更新，数据根据需要及时发生变化。数据仓库的数据主要供企业决策分析之用，所涉及的数据操作主要是数据查询，一旦某个数据进入数据仓库以后，一般情况下将被长期保留，也就是数据仓库中一般有大量的

查询操作，但修改和删除操作很少。通常只需要定期的加载、刷新。

（3）集成的。面向事务处理的操作型数据库通常与某些特定的应用相关，数据库之间相互独立，并且往往是异构的。而数据仓库中的数据是在对原有分散的数据库数据进行抽取、清理的基础上经过系统加工、汇总和整理得到的，必须消除原数据中的不一致性，以保证数据仓库内的信息是关于整个企业的一致的全局信息。

（4）面向主题。操作型数据库的数据组织是面向事务处理的，各个业务系统之间各自分离。而数据仓库中的数据是按一定的主题域进行组织的。主题是一个抽象的概念，是指用户使用数据仓库进行决策时所关心的重点方面，如客户、分销商、产品、活动等。一个主题通常与多个操作型信息系统相关。基于主题组织的数据被划分为各自独立的领域，每个领域有自己的逻辑内涵，互不交叉。

8.3.2 数据仓库的基本体系结构

数据仓库系统一般是由数据源、数据仓库管理、OLAP 服务器和前端工具四大部分组成的。在数据仓库体系中，数据的流程从后台处理开始，经过中间的存储管理，以前台的用户访问工具结束，其他如数据仓库管理工具、安全系统、元数据等则贯穿整个流程。其体系结构具体如图 8－1 所示。

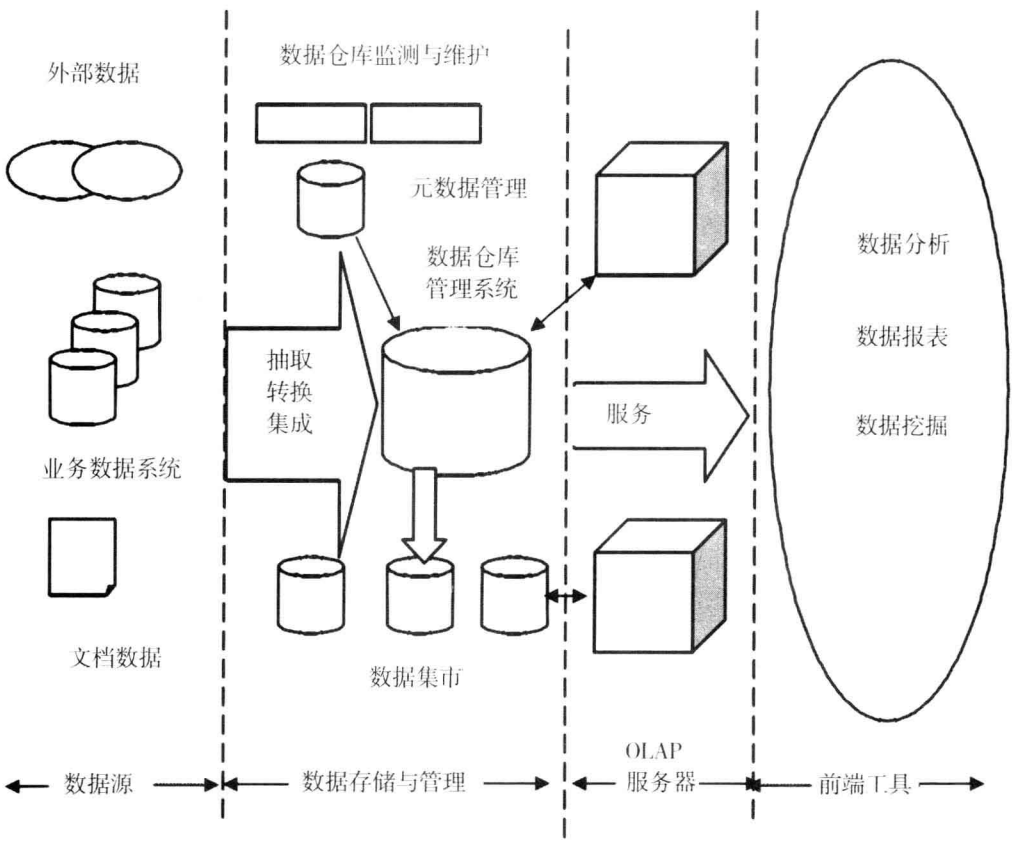


图 8－1　数据仓库的体系结构

数据仓库主体结构为数据仓库系统的基础，是整个系统的数据资源，通常是内部信息

和外部企业信息。内部信息主要是数据仓库系统中各种业务数据及各种文档数据。企业外部信息是各种法律法规、相关市场信息以及一些行业信息等。数据仓库中数据的存储和管理为数据仓库系统的核心。数据仓库实现的关键就是数据的存储、管理和分析，进而作出决策。数据仓库的组织管理方式是区别于传统数据库的，同时也呈现了有别于传统数据的展现形式。决定采用何种技术来作为数据仓库核心，需要从数据仓库的使用方式着手分析。目前现有的数据处理，主要包括抽取、清理、集成，提取有关信息。数据仓库按其数据的规模分为企业级别的数据仓库与部门级别的数据仓库。OLAP 数据服务器：对将要分析的数据进行有效集成及合并，按多纵向模型予以组织，进行多方位、多指标分析，得出相关结论。具体的实现方式主要有：MOLAP、HOLAP 和 ROLAP 等。MOLAP 基本数据和聚合数据均存放于多维数据库中；HOLAP 基本数据存放于 RDBMS 之中，聚合数据存放于多维数据库中。ROLAP 基本数据和聚合数据均存放在 RDBMS 之中前端工具：主要是一些报表工具、查询工具、数据分析等相关工具、数据挖掘工具以及各种数据分析与数据统计应用分析工具。数据分析工具对 OLAP 服务器进行信息交换，报表工具、数据挖掘工具对数据仓库中的大量数据进行提取，发现有用信息。

8.3.3 数据集市

原始数据从数据仓库流入不同的部门中以支持这些部门的定制化使用。这些部门级的数据库被称为数据集市，一个数据集市就是一个部门的数据体。数据集市可以看做整个企业数据的一个子集，包括特定业务单元、部门或用户集的值。该子集包含从事事务处理或企业仓库获取的历史数据、汇总的数据，并可能会有一些详细数据。数据集市是根据特定主题而不是根据数据集市数据库的大小来定义的。数据集市通常服务于单个部门或企业的部分用户，满足部门级用户的需求，因此数据集市也被称为部门级数据仓库。

在数据的粒度方面，数据仓库或业务数据库中的数据一般处于非常小的粒度级别，而数据集市中的数据则是处于概括级别。存储不同部门的数据集市包含了数据仓库或业务数据库中相同详细的数据的不同组合和选择。

有两种不同类型的数据集市，一种是独立数据集市，另一种是从属数据集市。独立数据集市的数据直接来源于业务系统。许多企业在规划数据仓库建设时，往往出于成本等各个方面的考虑，最后建成的就是这种独立的数据集市，用来解决某个部门的比较迫切的任务。从属数据集市的数据直接来源于数据仓库，一般是为那些访问数据仓库十分频繁的关键业务部门建立的，从而可以保证部门得到较快的响应速度，由于从属数据集市是在统一的数据仓库架构下开发的，因而可以保证得到较好的数据一致性。

无论数据集市提供何种功能，它们都必须被设计为主数据仓库的组件，以使数据的组织、格式和架构在整个数据仓库内保持一致。表的设计、更新机制或维度的层次结构如果不一致，可能会使数据无法在整个数据仓库内重新使用，并可能导致由相同的数据生成不一致报表的情况。例如，由销售部门的数据集市（该数据集市按地区组织销售人员）生成的汇总报表可能与由财务部门的数据集市（该数据集市按管理层次组织销售人员）生成的汇总报表不一致，但是也没有必要为获得一致性而将一个数据视图强加在所有的数据集市上，所以在数据集市设计时，通常设计一个一致的架构和数据格式，使得可以有很多不同的数据视图同时保持互操作型。从数据仓库组件的角度上去设计数据集市，无需考虑它们各自的功能和构造，这是数据集市设计的一个根本原则。

8.3.4 数据仓库的设计步骤

数据仓库的开发和传统的数据库开发有很大的区别。数据仓库开发的最初阶段往往出现一种死锁现象。开发者像他们在通常的开发过程中所做的那样，想从用户了解详细的系统需求。但另一方面，数据仓库的需求对用户来说在一开始并不明确，用户只能在看到数据仓库可能为他们做什么后，才能提出更具体的业务需求和建议。因此，数据仓库的开发应坚持自底向上的螺旋式开发策略，首先要探索数据仓库可能为用户做什么，并通过向用户演示，获得反馈后，进行进一步数据仓库开发，以此循环，逐步达到完善。图 8 - 2 显示的正是这种数据仓库的开发步骤。

从图 8 - 2 可以看到，数据仓库开发是从项目规划开始的。项目规划关注的是数据仓库项目定义和范围，其中包括项目准备评估和业务合理性证明。此外，项目规划将会集中关注资源技能熟练的人员的需求，同时还包括项目任务的分配、持续时间以及先后顺序等。

历史数据也就是业务数据库，它是数据仓库的数据来源，没有历史数据，数据仓库的开发无从谈起。因此，数据仓库开发必然要以历史数据为基础，即以数据为驱动。但是，数据仓库项目失败的经验告诉我们，以纯数据驱动的项目很容易遭遇失败，数据仓库项目必须要结合用户的需求，即用户驱动。数据仓库开发必须坚持用户需求驱动与数据驱动相结合的设计理念。从技术的观点来看，数据仓库开发团队的一个长期目标是构建企业信息基础结构，但是，这个目标往往很难达到，一方面是因为数据仓库开发的特点；另一方面是因为开发人员很难在项目的开始就具备企业级的观点。因此，在项目开发前夕，进行需求分析，明确数据仓库分析主题和其核心业务价值，选择数据仓库的开发策略，选择优先级高、开发难度相对低的主题作为数据仓库开发的对象。在明确开发主题和需求的基础上，设计维度模型，选择合适的存储方法和存储结构，对源数据库中的数据进行抽取、转换和加载。完成部分数据仓库测试，并获得用户反馈后，进一步明确需求。完善数据仓库的设计。

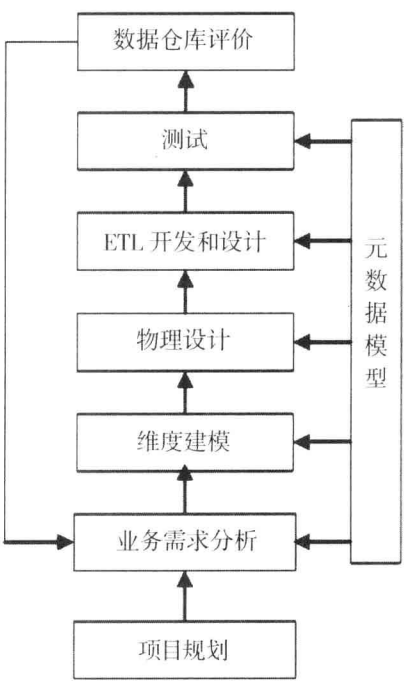


图 8 - 2　数据仓库的设计步骤

元数据对于数据仓库设计、开发和运作是至关重要的，尤其是在数据获取、转换和存取方面。元数据描述从操作型系统到分析型系统的映像，描述数据源、数据更新、总计数据的算法和数据提取频率。尽管面临着信息系统纷乱复杂、千头万绪且信息源层出不穷的局面，但是只要抓住了元数据，就等于抓住了关键，由于元数据涉及数据库及应用的核心和本质问题，因此，在数据仓库的开发过程中，元数据是在对企业商业智能需求的分析阶段就应该设计并一直贯穿于数据仓库应用全程的重要部分。

8.3.5 联机分析处理（OLAP）

（1）OLAP 的概念及数据组织

联机分析处理（On - Line Analysis Processing，OLAP）是使分析人员、管理人员或执行人员能够从多种角度对从原始数据中转化出来的、能够真正为用户所理解的并真实反映企业维特性的信息进行快速、一致、交互地存取，从而获得对数据的更深入了解的一类软件技术——OLAP 委员会的定义。OLAP 的目标：是满足决策支持或多维环境特定的查询和报表需求，它的技术核心是“维”这个概念。因此 OLAP 也可以说是多维数据分析工具的集合。

建立 OLAP 的基础是多维数据模型，多维数据模型的存储可以有多种不同的形式。MOLAP 和 ROLAP 是 OLAP 的两种具体形式。其中，MOLAP 是基于多维数据库存储方式建立的 OLAP；ROLAP 是基于关系数据库存储方式建立的 OLAP。

MOLAP 利用一个专有的多维数据库来存储 OLAP 分析所需的数据，数据以多维方式存储，并以多维视图方式显示。二维数据很容易理解，当维数扩展到三维甚至更多维时，多维数据库将形成类似于“超立方体”一样的结构。实际上，多维数据库是由许多经压缩的、类似于数组的对象构成的，这种对象通常带有高度压缩的索引及指针结构。MOLAP 结构的优势在于能够迅速地响应决策分析人员的分析请求并以最快速度将分析结果返回给用户，但在 MOLAP 结构中，OLAP 服务器的缺点就是通过读取预先处理的数据来完成各种分析操作，而这些预先处理操作是预先定义好的，这就限制了 MOLAP 结构的灵活性。

ROLAP 以关系数据库为核心，用关系数据库中的二维表来组织数据，表达多维概念。ROLAP 的数据仓库系统采用关系数据库系统，当数据仓库的数据模型确定之后，分散在企业各 OLTP 数据库中的数据被载入数据仓库并按数据模型的要求进行预处理。在 ROLAP 结构中数据的预处理程度一般较低（如果预处理程度太高，数据冗余量大，将使管理和维护更加复杂）。ROLAP 的主要特点是它的灵活性强，用户可以动态定义统计或计算方式。ROLAP 的缺点是它对用户的分析请求处理的时间要比 MOLAP 长。

（2）OLAP 的多维分析操作

维是人们观察数据的角度。如果企业需要观察从时间角度出发的产品销售的变化情况，那时间就是一个维（时间维）。同样，如果企业还关心在不同地区销售的分布情况，那地理分布也是一个维（地理维）。以某个维观察数据还可以存在细节程度不同的多个描述方面，这多个描述方面称为维的层次。一个维可以有多个层次。如时间维，可以从日期、月份、季度和年等不同细节层次来描述，因而日期、月份、季度和年就是维的层次。地理维中，城市、地区、国家、洲也都是层次。维的一个取值称为该维的一个维成员。如果一个维具有多个层次，那么该维的维成员是在不同维层次的取值的组合。例如，我们考虑时间维，其中含有年、月、日三个层次。在日、月、年上各取一个值组合起来，就得到

了时间维的一个维成员，如“2005 年 6 月 1 日”。

多维分析是指对以多维形式组织起来的数据采取切片、切块、旋转等各种分析，以求剖析数据，使最终用户能从多个角度、多侧面地观察数据库中的数据，从而深入地了解包含在数据中的信息、内涵。

（1）切片（Slice）

选定多维数组的一个二维子集的动作叫做切片，即选定多维数组（维 1，维 2，……，维 N，变量）中的两个维：维 I 和维 J。在这两个维上取某一区间或任意维成员，而将其余的维都取定一个维成员，则得到的就是多维数组在维 I 和维 J 上的一个二维子集，称这个二维子集为多维数组在维 I 和维 J 上的一个切片，表示为：（维 I，维 J，变量）。如图 8－3所示的切片。

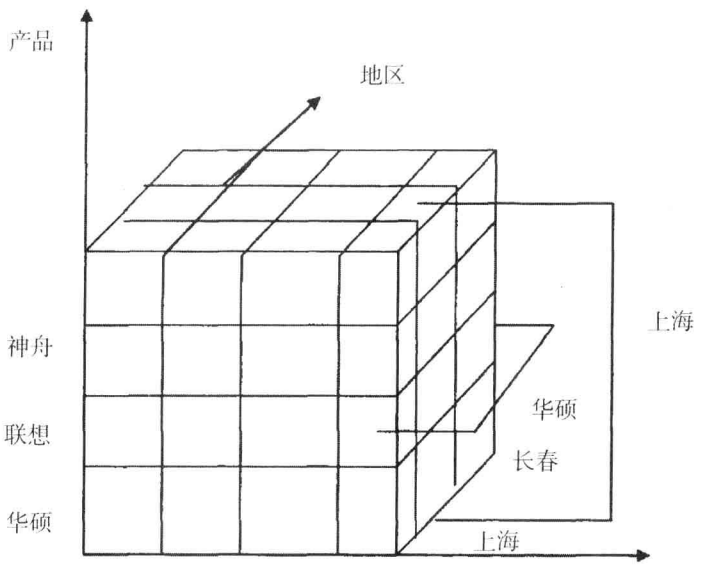


图 8－3　切片图

（2）切块（Dice）

选定多维数组的一个三维子集的动作称为切块。即选定多维数组（维 1，维 2，……，维 N，变量）中的三个维：维 I、维 J、维 R。在这三个维上取某一区间或任意的维成员，而将其余的维都取定一个维成员，则得到的就是多维数组在维 I、维 J、维 R 上一个三维子集，称三维子集为多维数组在维 I、维 J、维 R 上的一个切块，表示为：（维 I，维 J，维 R，变量）。切块与切片的作用与目的是相似的。从另一个角度来讲，切块可以看成是在切片的基础上，进一步取定各个维成员的区间得到的片段体，也即由多个切片叠合起来。

（3）钻取（Drill）

维度是具有层次性的，如时间维可能由年、月、日构成，维度的层次实际上反映了数据的综合程度。维度层次越高，细节越少，数据量少；维度层次越低，则代表的数据综合度越低，细节越充分，数据量越大。钻取包含向下钻取（Drill－down）和向上钻取（Drill－up）操作，钻取的深度与维所划分的层次相对应。向下钻取就是从较高的维度层次下降到较低的维度层次上来观察多维数据。反之，则执行的操作就是向上钻取。

（4）旋转（Rotate）

旋转即改变一个报告或页面显示的维方向。例如，旋转可能包含了交换行和列，或是把某一个行维移到列维中去，或是把页面显示中的一个维和页面外的维进行交换，令其成为新的行或列中的一个。

8.3.6 数据仓库应用实例

在企业管理中，经理人员和决策分析人员总是希望能够随时随地地访问到任何他们需要的信息。这就要求有一个体系结构来容纳各种格式的内部数据和外部数据，例如经营数据、历史数据、现行数据以及来自 Internet 服务提供商（ISP）的数据，此外还应该包含易于访问的元数据。今天的企业要求能够访问并综合来自各种数据来源的数据，能够通过充分地挖掘现有的数据资源捕捉、分析和沟通信息，进行复杂的数据分析，创建能代表业务分析员对数据看法的数据视图，通过概括、细化展示、多层次多视角地查看跨主题和跨业务范围的信息，发现许多过去缺乏认识或未被认识的数据关系，帮助企业管理者做出更好的决策。数据仓库技术、数据挖掘技术以及 OLAP 技术正是为了满足经历人员和决策分析人员的要求产生的。基于数据仓库的决策支持系统是这几种技术的综合体，它较好地集成了数据仓库、数据挖掘、OLAP 等多种信息处理技术，是辅助企业管理者做出正确决策的理想系统。

（1）电力客户数据仓库主题分析

对于传统客户关系管理来说，其分析内容主要包含以下七个方面，即：

①客户概况分析（Profiling）包括客户的层次、风险、爱好和习惯等；

②客户忠诚度分析（Persistency）指客户对某个产品或商业机构的忠实程度、持久性、变动情况等；

③客户利润分析（Profitability）指不同的客户所消费的产品的边际利润等；

④客户性能分析（Performance）指不同的客户所消费的产品按种类、渠道、销售地点等指标划分的销售额；

⑤客户未来分析（Prospecting）包含客户数量、类别等情况的未来发展趋势、争取客户的手段等；

⑥客户产品分析（Product）包括产品设计、关联性、供应链等；

⑦客户促销分析（Promotion）包括广告、宣传等促销活动的分析。

依据上述七个方面的内容并结合电力企业自身特点而确立的系统应用功能，据此可确定数据仓库的主题，如表 8－1 所示。

表 8－1 客户分析相关主题

一级主题	二级主题
客户基本情况分析	客户及容量统计、客户容量变化分析、不同均价的客户分布统计、客户收益分析等
区域客户分析	各个行业客户构成分析、各个电压等级客户构成分析、客户构成变化对用电量及电费受益的贡献等
客户发展潜力分析	能源消费结构分析、客户用电成本分析、用电设备分析、节能措施推广效果分析、客户赢利能力分析

（续表）

一级主题	二级主题
客户服务质量分析	客户意见调查、投诉举报统计、违章用电统计、停电影响分析、服务响应时间分析等
客户服务成本分析	线路改造成本计算、受电设备改造成本计算、客户服务人员工作量及成本计算
重点欠费客户分析	客户信誉度分析、欠费历史统计、回收难度分析、欠费性质分析等
客户服务需求分析	客户负荷预测分析、客户服务需求分析、客户满意度分析、客户购买趋势分析等

（2）数据仓库数据模型及实现

本系统以 Oracle10g 数据库为数据仓库引擎。数据仓库以星型模型组织数据实现多维数据仓库。如图 8－4 所示（简化），中间的表为事实表，其他是维表。

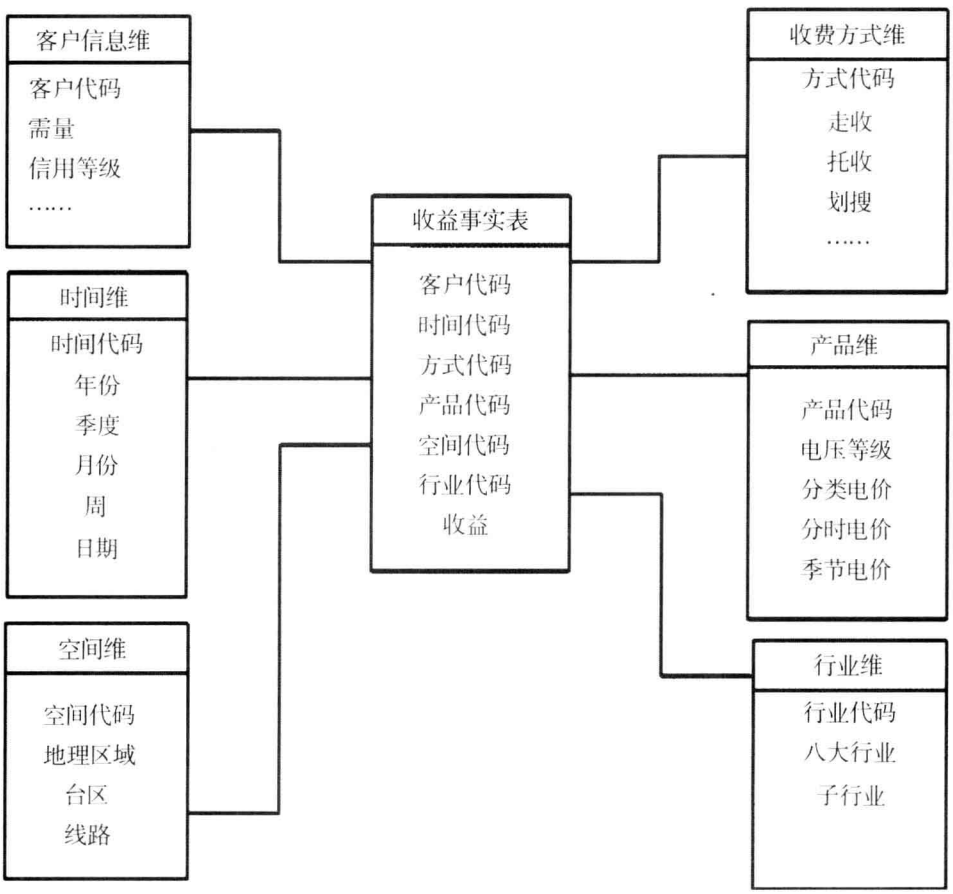


图 8－4　售电分析数据存储结构

事实表有多个连接，分别连接到各个维表上；而维表之间是不连接的。事实表使用前缀 fct 命名包含有进行售电分析的所有指标，如收益分析度量数据包括了收益分析所涉及的各项属性指标如售电量、售电收入等。该表主键由各维表的代码组成，它是一个很大的

关系表，要存储进行查询的各类信息。维度表维表使用前缀 dmn 命名，描述本维的类别代码，各层次的层次代码，通过主键与事实表相连。其中的维包括时间维、地区维、用电行业维等，其中每个维又具有各自的层次结构。粒度表存放每个维的每层粒度代码和名称。如八大行业是行业维的类别，八大行业的粒度表存放每个行业的代码和名称。通过主键，使指标表、维表、粒度表联系在一起，这样就用二维关系表达了多维的概念。通过这些联系，运用数据仓库系统的查询语言，就可以进行多维查询、多层面的分析。

事实表的特征是：①主要是数值型可加数据；②多个不同外键（连接到维表中）；③复合关键主键；④对用户的只读数据；⑤大的数据容量（有上亿行）。维表包含了完整的商业维的定义。维表中的关键字是唯一的。每一个维表都有许多属性，如说明、地区和设备等，都是为了完整描述一个维。维表的主键和事实表之间有个外键相连。这个结构支持复杂的查询和下拉式的分析。维可以看成是用来组织仓库中数据的种类，这些维还可以是有层次级别的。维表不像事实表那样有很多行，而且维表中的数据可以改变，比如一个客户的地址和电话号码可能改变。因此维表容许更新。维表的特征是：①以字符型数据为主；②多个列用来管理维的层次；③一个主键；④可更新的数据；⑤较少的数据量。

（3）数据仓库关键技术及实现

数据仓库并没有严格的数据理论基础，也没有成熟的基本模式；且更偏重于工程，具有强烈的工程性。因此，在技术上人们习惯于从工作过程的方面来分析，并按其关键技术部分分为数据的抽取、存储与管理、数据的表现以及数据仓库设计的技术咨询四个基本方面。

a. 数据的抽取

数据的抽取就是数据进入仓库的入口。由于数据仓库是一个独立的数据环境，它需要通过抽取过程将数据从联机事务处理系统、外部数据源、脱机的数据存储介质中导入数据仓库。数据抽取在技术上主要涉及互联、复制、增量、转换、调度和监控等方面。数据仓库中的数据并不要求与联机事务处理系统保持实时的同步，因此数据抽取可以定时进行，但是多个抽取操作执行的时间、相互的顺序、成败对数据仓库中的信息的有效性则至关重要。

b. 存储和管理

存储和管理：数据仓库的真正关键是数据的存储管理。数据仓库的组织管理方式决定了他有别于传统的数据库，同时也决定了其的表现形式。数据仓库需要解决的四个问题，即对大量数据的存储和管理、并行处理、针对决策支持查询的优化以及支持多维分析的查询模式。为此，需要使用数据库分区技术将容量很大的事实表分解成为若干个小型表，如：按时间范围分区，则可实现以时间查询的优化。利用分区和可传输表空间复制、转移表空间集，完成数据仓库的填充。

c. 数据的表现

数据的表现：数据表现相当于数据仓库的门面，其性能主要集中在多维分析、数理统计和数据挖掘上。而多维分析又是数据仓库的重要表现形式，近几年来由于互联网的发展，使得多维分析领域的工具和产品更加注重提供基于 Web 的前端联机分析界面，而不仅仅是在网上发布数据。

数据仓库的重要功能在于支持 OLAP，从而支持决策功能。而 OLAP 通常涉及大量的数据，并且要求快速地响应。数据仓库的解决方案是建立实视图（Materialized View）。它

与数据库的“视图”概念不同之处在于：它不是虚拟的，而是已经计算、含有大量数据并存储在数据仓库中的表。它是对源数据分析之后，针对分析主题对相关原始数据进行连接、投影、汇总而得。这样，在进行决策分析时，访问的是数据仓库中的主题数据，即与主题相对应的实视图，而不是原始数据。以营销决策支持系统中客户分析这一主题为例，建立实视图——客户分析表，它主要包含时间、客户、销售量、销售额、销售渠道等数据，通过对发票数据、合同数据、客户数据等源数据进行分析连接、投影、汇总而得。针对每一主题，都要建立实视图，复杂的分析主题可能对应多个实视图。实视图是提高系统响应时间的一个关键技术。

（4）基于数据仓库的数据挖掘及实现

数据挖掘是基于数据仓库的分析型客户关系管理的核心部分。不同企业应用不同，环境不同自身情况不同，因而挖掘方法也不相同。优秀的数据挖掘系统应该具有功能完备性、可扩展性、多数据模型适应性和可视化表现能力。基于数据仓库的电力企业客户关系管理系统所运用的分析方法应该是统计方法和数据挖掘方法的结合，如此才能在电力企业内部形成比较易于实践的评价分析体系。其中客户行为分析包括：信用度分析、贡献度分析、客户赢利能力分析、满意度分析、客户欠费分析、客户用电异常分析等；围绕客户的市场分析包括：客户购买趋势分析、客户需求分析、区域客户结构分析等。其中客户细分算法可采用模糊聚类、k－中心聚类算法、决策树 C4.5 算法；信用度评价方法可用模糊聚类、Logit 统计分析方法；电能欠费分析宜采用决策树算法；客户贡献度、客户赢利能力分析、满意度分析属于客户价值分析宜用统计预测方法。购买趋势分析属于数据挖掘中趋势挖掘的范围，宜用神经网络、模糊聚类方法。

第 9 章　数据挖掘在信用评估模型中的应用

9.1　我国商业银行运用数据挖掘技术的必要性

近年来我国银行业个人信贷业务蓬勃发展，不仅为消费者带来极大的便利，对金融机构也有积极的经济意义。个人信贷品种亦呈现多元化发展，从种类上看，包括住房贷款、汽车贷款、助学贷款、信用卡贷款等；从信贷方式上看，有抵押、质押、担保和信用等多种信贷方式；从用途上看，主要用于个人消费需求。从商业银行的角度来说，个人消费信贷由于其风险较工商企业生产经营类贷款分散，数额也相对较小，不良贷款率较低，因此被认为是优质资产，从而成为商业银行的优先发展对象。有些银行甚至把个人贷款作为主打产品，把自己打造成零售银行。但是金融市场向来都是风险与收益对称，随着我国个人消费贷款业务的广泛开展，金融机构面临的各种风险也逐渐显露出来，其中最值得关注的是信用风险，这也是现代市场经济道德秩序的核心。信用风险表现在消费信贷违约情况频繁发生，其带来的不良贷款使银行蒙受很大损失。信用风险既影响银行的经营活动效益，又影响金融运行的稳定性和经济发展的持续性，更是传统社会向市场经济社会转型过程中必然要面临和解决的一个问题。国际巴塞尔资本协议中列出了一系列指导文件，将银行降低风险的重要性提到了更高的层次，并提供了多种度量风险的方法。虽然我们不可能消灭风险，但我们必须以科学的手段来准确地评估风险、以正确的策略来有效地控制风险。个人信用是整个社会信用体系的核心和基石，是社会有序发展和经济持续健康增长的前提，这一点从美国次贷风波带来的巨大损失中也可见一斑。在商业银行等金融机构大力推行个人信用贷款的同时，如何建立一套严格准确的消费贷款个人资信评估衡量制度，对于金融机构提高个人资信评估技术与管理能力至关重要。

最近两年，有些发达地区的商业银行开始采用信用评分系统，来进行消费信贷的审批。这些银行的信用评分一般是借鉴国外商业银行的信用评分体系，并结合我国实际情况进行调整。但这些银行使用信用评分的范围和程度都十分有限，而且在项目的选取、分值的确定等方面，人为因素较多，影响了信用评分的客观性和公正性。我国商业银行应尽快引进国外银行业先进的风险管理理念和技术，掌握实用的信用评分方法，并根据所处的社会经济环境和自身的经营状况加强消费信贷借款人信用风险的研究，提高信用风险的管理能力，破除制约消费信贷业务发展的“瓶颈”。

但是，国外银行业近年来对数据仓库建设的投入却正在日渐加大，数据挖掘技术的应用正在逐渐变得普遍和重要，根据美国 TheTowerGroup 咨询公司的有关调查资料：1998 年

全球500家大银行中，已经有近90家建立了数据仓库并使用数据挖掘技术为其服务。据了解，在这500强银行中，目前至少有300家银行已经从事数据挖掘工作了。

与国外商业银行相比，目前我国商业银行在数据管理与数据分析方面还存在很多问题，主要是以下几个方面：

①缺乏以国内环境为背景的范例和成功应用。现在看到的所有案例和应用，大多是从国外引入的，由于存在着文化差异，这些并不是非常适合国内的实际情况。

②数据仓库虽然已经部分得到建立，但数据的不完整性尤其突出。

③缺乏整体规划，各个业务管理系统都是自行设计开发的，缺少整体性，同时缺乏统一的信息管理标准和规范。例如会计部门与储蓄部门，它们有着完全不同的数据管理系统。

④信息量大，而现有系统不能够有效地运用已存在的信息，造成信息资源的严重浪费。几乎所有的分析全是基于传统技术上的统计分析，只能够提供简单的业务统计报表等统计功能。

对信息的浪费又造成人们对信息的不重视，越发忽视信息来源的可靠性、真实性和信息的完整性，这就形成了恶性循环，引发了又一轮的信息缺乏。这就要求用先进的分析技术去挖掘出有用的知识，使得已有信息能够得到充分地利用。而数据挖掘技术则是目前功能最强大的数据分析技术，它包括了传统统计技术，还有人工智能、遗传算法等等，是目前所能采取的最好的数据分析技术。同时，也应该意识到，采用数据挖掘技术是国内银行业必走的一步，它可以为商业银行带来很多直接效益：

①有助于在市场、质量、效益、成本、改革等方面提高和加深商业银行对业务的理解和认识，并将有效提高商业银行经营管理决策的科学性和管理效率。

②将推动商业银行对现有信息管理体系的重构，信息管理体系的整合，也将给商业银行的机构和管理体制的改革带来积极的影响。

③有效地连接商业银行各类业务管理系统间的信息交流，有效发挥现有各类业务管理系统的整体效益，并推动商业银行信息管理标准化工作的进程。

也就是说，进行数据挖掘是商业银行生存发展、提高竞争力的需要。如果能在商业银行中合理有效地使用数据挖掘技术，就可以从个人客户账户（信用卡、各类储蓄存单、卡折等）上发生的明细了解客户的收入来源、消费爱好及习惯、收支发生的频度等，分析其消费支付行为，对客户账户进行分类，根据明细数据给客户计分，采取措施优化客户服务，例如，给予优良客户消费优惠折扣以鼓励消费并吸引潜在客户。对于单位账户，可以通过了解客户经营的产品结构、行业性质、市场占有率、社会需求及国家的有关政策等，分析其发展趋势，确定其资金偿还能力、信用等级等，也可通过对一些潜在的客户数据分析，拓展有前途的客户，最终起到降低成本与风险、提高效益的作用。在这些方面目前已有一些成功的案例，如旧M在对信用卡欺诈的侦测、目标市场和固定资产评估等的数据库分析中成功地使用了这些算法和手段。拥有20亿美元资产的Frstar银行根据消费者是否有家庭财产贷款、赊账卡、存款证或其他储蓄、投资产品，将他们分成若干组，使用数据挖掘程序预测什么时间向每位客户提供哪种产品。

从上面的论述可以发现，银行如果能充分利用已有的数据资源，利用先进的数据分析技术挖掘出其中的信息黄金，深化和客户的关系，发现潜在的客户群，抢占市场，就将对降低投资风险、避免高风险投资和提高投资效益起到意想不到的作用。

在过去银行业没有进行数据大集中的时候，每个分支行都有各自的个人住房贷款数据库，每次上报的数据只是原始数据或是简单的统计汇总、平均、加减等，分析方法和分析工具落后，综合性信息管理和决策支持管理的功能很弱。由于受到当时处理数据手段水平、数据库容量以及计算机运行速度等等方面的限制，银行过去只能做一些简单的分析工作，为上层决策也只是提供表面的个人住房贷款数据，这就造成了银行管理人员在实际工作中，往往由于不能全面把握内部和外部的信息，缺乏信息的交流，从而对未来发展趋势做出错误判断，不能迅速地对个人住房贷款的风险做出评估，导致决策失误。

但目前我国对个人信用评价缺乏科学的方法。在对贷款人的信用风险进行评估以及决定是否发放贷款时，主要依靠授信机构的信贷人员进行主观判断，从而决定是否给予某个消费者一定的信用消费权利，精确的信用评估方法几乎没有使用。这种主观评价方法的缺点显而易见：首先主观性较强，不能精确地计量消费者的信用风险；其次在对风险不能精确计量的同时，风险控制的成本却较高。个人信贷业务的特点是单笔业务的交易量较小，但业务的数量却较多。因此主要依赖信贷人员判断的信用评估方法，不仅无法对个人信用程度进行精确的计量，而且无法有效地降低单笔贷款的管理成本。个人消费信贷的快速增长与中国商业银行落后的信用风险管理技术和手段形成的强烈反差也迫切地要求商业银行建立与消费信贷增长相适应的风险评估方法。

9.2 信用的基本概念

（1）信用

《中国大百科全书》（经济卷）对“信用”定义为：以偿还为条件的价值活动的特殊形式。在商品交换和货币流通存在的条件下，信用指的是债权人以有条件让渡的形式贷出或赊销商品，债务人则按约定的日期偿还借款或赊销商品，并支付利息。需要注意的是，它不仅指简单的信贷活动和信用关系，而且含有货币借贷者的偿债能力、履约记录、守信程度及由此形成的社会信誉等多种意义。

（2）个人信用

个人信用是建立在信用基础上的涉及个人的信用总和，是指社会根据公民现有和历史的记录，为其可以预见的偿付能力和偿付意愿而给予相应的信赖和评价，使该公民在经济生活中可以因为这种评价而便利地获得资金、物资等经济支持。可见，个人信用不仅仅指个人在商品借贷中的简单借与还的行为，还是一个人在经济、道德、社会、法律等包括个人与社会的各种因素的观念和能力方面的反映。它所包含的内容不仅有现在的，还有过去的；不仅包括反映其偿付能力的信息，例如个人基本情况、个人收入、个人资产等，还包括能够体现其偿付意愿的历史信息，例如个人信用卡消费记录、个人诉讼记录、个人贷款记录等社会信用信息。

（3）个人信用评估

个人信用评估就是通过综合考察影响个人及其家庭的内外客观、微观环境，包括经济、金融、司法、工商、财产等过程在内的，使用科学严谨的分析方法，对个人及其家庭的资产状况、履约各种经济承受能力和信誉度进行全面评判与估价，以一定的符号表明其信用状况。概括来说，个人信用评估是个人资产和个人信用状况的综合反映。它涉及个人

或者家庭的资产和本人的信用，包括个人的短期资产、长期资产、无形资产和其他资产，其结果无须向社会公布，只要贷款人或者需要掌握个人信用等级的机构了解即可。

（4）个人信用评估体系的优点

个人信用评估体系为信贷决策带来很多变革，其优点主要体现在以下几方面：

第一，节省信贷审批时间，提高审批效率。据美国商业银行理事会（BusinessBanking-Board）的一项研究显示，利用传统信贷审批方法对一个个人贷款的申请审批所花费的时间在12.5小时至两个星期之间，而利用信用评分体系时，所花费的时间可缩短到1个小时之内。

第二，减少贷款审批过程中的主观性。在传统的贷款审批过程中，不同的贷款审批人员对于申请人有不同的判断标准，贷款审批人对申请人的判断很大程度上受该决策者过去经验的影响。利用信用评估体系，可以保证贷款机构对不同的申请者使用的是同一个标准。

第三，信用评估体系使得在贷款机构进行信贷决策时可以考虑多种因素，而传统的主观判断所考虑的因素不可能很多。

9.3 信用风险的影响因素

分析信用风险首先要明确其产生的原因，总体来看，引发信用风险的因素主要包括各经济主体的经营状况、盈利水平、信用能力以及与这些因素相关的体制制度、治理结构和管理水平等微观因素，同时还包括一国的经济环境、政策制度等宏观因素。

对于商业银行的借款业务，影响其信用风险的因素主要有：

（1）借款人的信用程度，即借款人的信誉水平和履约意愿。信用程度与借款人的性质紧密相关，一般来说，政府部门的信用程度高于企业，而大型企业又高于中小企业。同时，具有自主经营权的企业会从自身利益出发，努力经营，信用程度较高。

（2）借款人的财务状况。客户的财务状况不仅包括其经营能力、获利水平，还包括其获得其他渠道资金的融资能力。一个拥有稳定现金流流入或灵活便利融资渠道的企业具有及时履约的财务能力。反之，如果一个企业经营不善，没有充足的现金流预期，且再融资能力低下，那么它将具有很大的违约风险。

（3）贷款的方式。贷款的方式主要有信用贷款、抵押贷款、票据贴现等形式。信用放款没有抵押物，单以借款人的信用作为保证，所以较有担保物的抵押贷款和真实交易作保障的票据贴现来说，信用贷款具有较高的信用风险。

（4）宏观经济状况与政策制度因素。整个社会宏观经济发展情况将影响到各经济主体的经营前景，从而对他们的还款能力产生影响。各国的实际经济运行证明，在宏观经济运行良好，经济处于繁荣的时期，商业银行的贷款业务面临的信用风险一般不高，形成的不良贷款数额也相对较低，而在经济低迷、市场交易不活跃的衰退时期，往往潜在着较高的信用风险，同时商业银行的大量不良贷款也会在这一时期形成。

9.4 风险评估的特点

（1）简洁性。信用评估以简洁的字母数字组合符号揭示评价对象的信用状况，是一种对研究对象进行价值判断的简明的工具。

（2）可比性。各信用评估机构的评级体系使同行业受评企业处于同样的标准之下，从而揭示受评企业在同行业中的信用地位。

（3）全面性。信用评级就受评企业的经营管理素质、财务结构、偿债能力、经营能力、经营效益、发展前景等方面全面揭示企业的发展状况，综合反映企业的整体状况。

（4）公正性。信用评级由专业的信用评级部门作出，秉承客观、独立的原则，较少受外来因素的干扰，能提供客观、公正的信用信息。

（5）形象性。信用评级是企业在资本市场的通行证，一个企业信用级别的高低，不但影响到其融资渠道、规模和成本，更反映了企业在社会上的形象和生存与发展的机会，是企业综合经济实力的反映，是企业在经济活动中的身份证。

（6）社会信用的基础性。当今社会的信用基础较为薄弱，通过信用评级，使社会逐步重视作为微观经济主体的企业的信用状况，从而带动个人、其他经济主体和政府的信用价值观的确立，进而建立起有效的社会信用管理体制。

9.5 信用评估模型的主要应用技术

（1）经济计量技术，包括 Logit 和 Probit 分析、现行和多元判别分析和多元回归等方法，通过这些方法构建违约概率和违约溢金模型。

（2）最优化模型。是一种数学规划技术，通过确定借款人和贷款属性的最佳权重以达到使借款人的损失降低到最小。

（3）神经网络，通过模仿人脑中内部神经元网络的最小决策单元，从而能够模仿人类大脑功能的预测模型。

（4）使用直接估算、估计和模拟技术的混合系统。

（5）基于规则和专家的技术，通过结构化方法模拟经验丰富的专家确定分析决策的分析过程。

（6）决策树，以已有的实例作为训练样本集，通过学习归纳得出计算模型。

9.6 基于数据挖掘的农户小额贷款信用评估模型研究

近年来，随着我国对“三农”问题的推进，为了使农村经济向着更好更快的良性发展，我国开展了以农村信用社参与为主的农户小额贷款业务。经过十多年的发展，我国农村信用社小额贷款在缓解农村金融货币市场的供求矛盾、拓宽农村信用社的资金运用渠道、优化农村金融结构等方面都取得了显著的经济效应。但是在进行小额贷款业务的过程

中又不得不面临贷款农户的信用风险问题。对于农村信用社来说，对农户是否发放小额贷款主要根据信贷审批人员的经验来决定，个人的主观因素对评估影响很大，而随着业务量的大幅增加，人员相对不足，势必会造成审批时间长、贷款效率低等问题，严重阻碍了信用社的小额贷款业务的长足发展；另一方面，由于贷款对象是一个特殊的群体，以及贷款对象经营的是一个低效、弱势的产业，受自然灾害的制约因素较多，各类拖欠贷款的现象较为频繁，不良信贷比率居高不下，这些都是银行业面对的不可避免的风险。因此，建立针对信用社小额贷款的信用评估模型，对于我国农村信用社的生存与发展，其意义就显得更加重大。

本节从我国农村信用社的信用风险评估的现状以及存在的问题分析出发，通过对目前国内外个人信用评估模型的研究和分析，同时结合农户小额贷款业务自身的特点，提出了基于数据挖掘中的分类算法建立农户信用风险评估模型。

小额贷款业务是20世纪80年代以来国际上发展起来的新兴事物，由于其灵活性和有效性的特点，在全球范围内越来越受到关注。近年来，小额贷款作为一种新的扶贫方式和金融创新，在我国的农村信用社等金融领域有了广泛而长足的发展。而在进行小额贷款业务的过程中有不得不面临贷款农户的信用风险问题。

对于农村信用社来说，为农户是否发放小额贷款往往要根据小额贷款业务的审批人员的以往经验来决定，因此导致个人的主观因素对信用评估的结果影响很大。而随着业务量的不断增加，加上信贷人员相对不足，势必会造成审批时间过长、贷款效率低等问题，严重阻碍了信用社的小额贷款业务的长足发展。此外，由于小额贷款的对象是一个特殊的群体，以及贷款对象经营的又是一个低效、弱势的产业，受自然灾害的制约因素较多，各类拖欠和逃废债务的现象频繁出现，不良信贷的比率呈逐年上升的趋势，这些都是银行业面对的不可避免的信用风险。因此，如何建立1个关于农户小额贷款的信用风险评估模型来降低农村信用社的信贷风险成为目前银行业所面临的紧迫问题。数据挖掘技术正是解决这一问题的最佳方法。将数据挖掘技术引入信用评估模型的构建研究中，能够在一定程度上解决目前银行业所面临的一些信贷风险问题。在国外利用数据挖掘的方式进行金融风险管理已经相当普遍，而在国内才刚刚起步，处在积极探索和经验积累阶段。农户小额贷款的信息系统的快速发展使其积累了大量的数据，这些数据涉及面较广，指标也很多，但是对这些数据进行各种深层次的分析、综合、提炼、挖掘和展现做的还不够深入，因此可供银行管理者和信贷人员利用的分析、预测数据不多，能辅助决策的有效信息就更少了。

9.6.1 我国农村信用社农户小额贷款面临的问题分析

农户的小额贷款是服务“三农”中重中之重的工作，但是农村信用社在对农户进行小额贷款面临着很多风险。主要包括以下两个方面：

（1）农户的信用风险。农户的小额贷款是面向广大农民发放的贷款，而农户的诚信度的高低很难去实行量化，所以这就对信贷人员提出了更高的要求，能够准确地把握成千上万农户的信誉程度，缺乏公正有效的个人信用评估手段，不仅阻碍了小额贷款业务的发展，同时也给农村信用社等金融机构的资产质量管理带来风险。

（2）农户生产经营风险。农户小额贷款的用途主要是用于种植业、养殖业、农户产品加工以及教育等消费性贷款，其中种植业受自然灾害和市场变动等因素影响较大，农民又是弱势群体，一旦出现自然灾害导致农业减产、农产品销售受阻，将直接导致还贷能力减

弱，从而造成贷款风险。

从以上两点我们可以看出，农村信用社在对农户发放小额贷款时，面临着很多关于农户信用风险的问题。因此，如何建立1个信用风险评估模型降低银行的信贷风险成为目前银行业所面临的紧迫问题。数据挖掘技术中的决策树分类算法正是解决这一问题的最佳方法。在国外利用数据挖掘的方式进行金融风险管理已经相当普遍，而在国内才刚刚起步，处在积极探索和经验积累阶段。农户小额贷款的信息系统的快速发展使其积累了大量的数据，这些数据涉及面较广，指标也很多，本文通过对这些数据进行更深层次的收集、加工、综合、提炼、挖掘和展现，从中提取农户小额贷款评估模型方面的关键性数据，并对农户小额贷款系统中的数据信息进行有效的数据挖掘，发现相应的知识，进而得到农户小额贷款信用评估模型，从而为小额贷款的风险管理提供决策依据。

9.6.2 信用风险评估模型风险影响因素的确定

根据数据挖掘的基本过程，在进行数据挖掘之前，首先的任务是理解实际业务问题，在业务需求详细描述的基础之上提出问题，根据业务需求获取原始数据，从而建立相应的数据集合。本章将根据农村信用社农户小额贷款信用风险评估的相关规定和业务需求明确挖掘目标，选择合理的贷款风险影响因素，进而建立农户小额贷款风险评估体系。

虽然不同层次的人员对信用风险评估的着眼点有所不同，考虑农户的情况也有些差异，但是大致方向是相同的。通过对《黑龙江省农村信用社小额贷款实施细则》以及《农村信用社小额贷款申请书》的分析，并结合与信贷人员交流，以及对金融专家的更深入的探讨，经过综合考虑，将影响借款农户的还贷情况的风险因素主要分成：即借款人基本信息、借款人收入及负债情况信息、借款人的个人征信信息、贷款产品基本信息、贷款用途信息。本人对这些风险因素的不同类分别选取了具有代表性的因素，如下所述：

（1）借款农户的基本信息

借款农户的基本信息主要是描述借款农户自身属性的信息。这些基本信息是用来考察借款农户在履行合同的过程中是否违约的重要考虑因素之一，主要包括借款农户的年龄、性别、文化程度、婚姻状况、家庭人口状况，住房状况。借款农户的基本信息能够从一个侧面体现借款农户的还款能力，并且这些信息是容易从信息系统中提取得到的。

（2）借款农户的收入及负债状况信息

借款农户的财产状况、负债以及支出状况信息是能够比较明确地反映出借款农户的还贷能力的，也是个人信用评估体系中比较重要的指标，也是农村信用社的小额贷款信用风险是否发生的关键因素。主要包括：农户的月平均收入、借款农户月偿还该笔小额贷款占家庭月总收入的比重。

（3）借款农户的个人征信信息

借款农户的个人征信信息主要是指借款的农户在农村信用社征信报告中所记录的借款农户两年内的历史还款行为记录，主要包括借款农户是否有违约债务的历史。借款农户的个人征信信息是考察借款农户历史还款信用的重要信息，也能够在一定程度上反映借款农户的大的素质以及对所借款项的偿还意愿。

（4）贷款产品的基本信息及贷款用途信息

贷款产品的基本信息是影响信用风险评估的主要指标，主要包括贷款期限、贷款金额（小额贷款金额相对较少）。贷款用途主要是要考虑农户将款项用于哪方面的投资，这种投

资方向也是影响农户能够按时还贷的重要指标，如表9－1所示。

表9－1　农户小额贷款信用风险评估模型影响因素表

类别	名称	取值定义
农户基本信息	年龄	年龄值的区间表示
	性别	男、女 两种值
	教育程度	文盲、小学、初中、高中、中专、大专、大学本科、研究生及以上
	婚姻状况	未婚、离异、已婚
	家庭人口状况	人数≤3 3＜人数≤5 人数＞5
	住房状况	有、无住房
收入及负债基本信息	月平均收入	月平均收入额
	月偿还该笔小额贷款占家庭月总收入的比重	月还款额/月平均收入额
个人征信信息	是否有违约债务历史	有违约债务历史 无违约债务历史
贷款产品基本信息及贷款用途信息	贷款期限	1年、1～2年、2～3年
	年贷款金额	金额≤1万元 1万元＜金额≤3万元 金额＞3万元
	贷款用途	纯农业生产 种植、养殖业、林果业 农副产品加工、农村流通业

确定农户信用风险评估模型中风险影响因素后，就可以应用数据挖掘中的决策树算法选择其中对农户信用风险水平影响较大的因素建立信用评估模型。

9.6.3　农户小额贷款数据挖掘目标

本节研究的目标是通过决策树算法对农村信用社以往对农户进行小额贷款的数据库中的大量信息进行处理分析，以求达到对农户小额贷款进行风险评估和防范不良贷款产生的风险，尽量避免风险的发生的目的。其最终目标包括：

（1）得出借款农户的信用等级分类规则：根据信用评估模型得到的借款农户的信用评估，进一步得到对不同分支的借款农户的信用等级的分类规则。本文的建模目标只是将农户划分为两类：好客户；坏客户。

（2）预测借款农户的信用状况：通过对训练样本集的学习提取出关于这些借款农户数据的特征，当有新的借款农户输入时就可以较为准确地确定该借款农户的信用等级。

（3）建立风险评估模型：根据借款农户的信用状况和一些其他的特征综合评价该农户的信用风险等级。从而可以防止不良贷款的发生。

本节对借款农户的信用情况进行了比较全面的研究，并结合小额贷款的一些特征建立

了信用风险评估模型，在此基础上得到的小额贷款信用风险大的分类等级规则，从而能够对新得到的预测模型进一步对新的借款农户的还贷情况作出较为准确的预测。

9.6.4 数据采集及预处理

（1）数据采集

在对农户小额贷款的数据进行采集的过程中，要做到采集的数据完整、不遗漏业务目标所涉及的各个环节，也就是说采集的数据字段中要包含农户小额贷款风险评估模型中所涉及的各个影响因素。同时也要求各个属性的数据要按照一定的规则有机地、准确地衔接起来。这些通过数据采集而重新组织的数据，要以较容易读取的数据结构方式存储，并能够全面地描述业务目标。根据本文信用风险评估模型的影响因素以及数据挖掘目标，经过分析得到与本研究有关的数据表，如表9-2所示。

表9-2 数据采集所选择的数据表

数据表名称	说明	与数据挖掘有关的属性
农户贷款申请表	贷款的金额情况及审批情况	贷款金额、贷款期限、月还款额、贷款日期等
借款农户基本情况表	借款农户的基本信息	姓名、性别、年龄、出生日期、身份证号、婚否、学历、健康状况
家庭经济情况调查表	有关农户的经济情况调查	月平均收入、家庭人数
还款信息表	记录每次还款的情况	还款金额是否有违约债务

除了要确定建立评估模型所用到的数据表之外，还要确定采集数据的时间问题，由于数据库中的表每天都会发生变化，因此本文采集数据选用的是某一农村信用社固定的一天业务结束后的数据库。并且要选取数据分析的表现期，即农户在贷款后1~2年的违约暴露情况，由于小额贷款的特点一般时间都较短，所以本文的农户信用风险评估模型中的模型表现期选取一年。本研究通过随机序列发生器，从农户小额贷款数据中随机抽取一半的数据作为研究的样本数据，另外一半用于对模型进行测试评估。

在进行数据挖掘过程中，大量的数据和多维的数据使得数据挖掘的实现遇到很多困难。这就需要通过采用适合于农户小额贷款数据挖掘需求的方法，对数据库中已经存在的数据进行降维处理，从中取得适合挖掘需求的数据字段。数据挖掘算法通常是对二维关系表进行的，在本文的研究应用中，通过SQL查询语言定义几个基本的数据抽取方法：

①多表合并抽取方法

在小额贷款数据库管理系统中，关于农户贷款信息的数据都是通过多张的二维表来表达的，数据表之间存在一对一、一对多、多对多的对应关系。先对表进行并运算成为一张新的二维表。由于多对多的关系可以转换为一对多的关系，因此仅需设计一对多数据表的抽取方法。多表合并的数据抽取方法如图9-1所示。

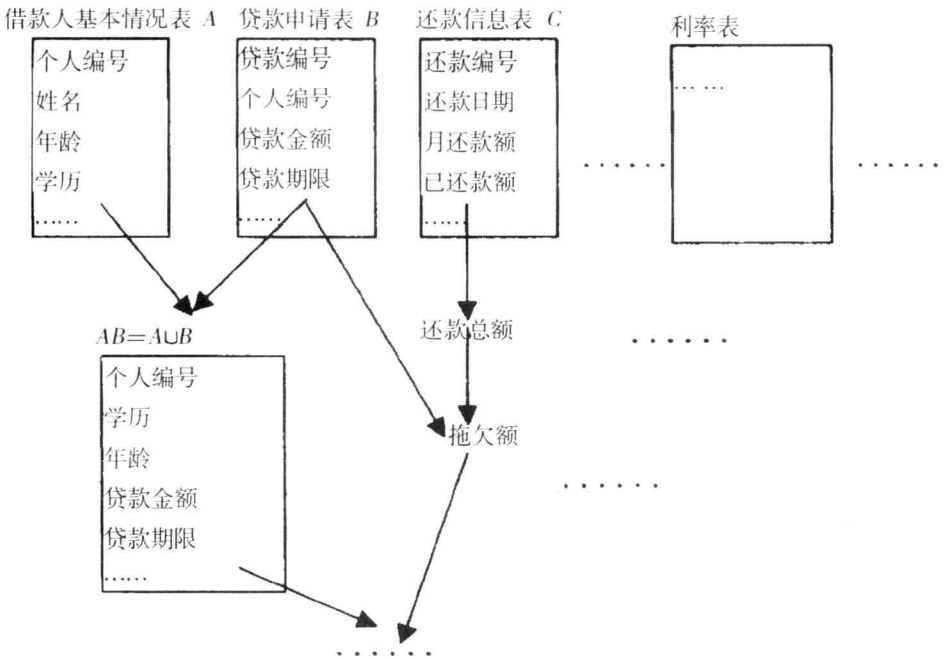


图 9 - 1　数据抽取方法示例

对于图 9 - 1 中所示的借款农户的基本情况表（*A*）和贷款申请表（*B*），抽取这两张二维表中对本次数据挖掘有用的信息，如 *A* 表中的“年龄”等信息，*B* 表中的“贷款金额”、“贷款期限”等信息，最后得到合并后的 *AB* 表。

②分组统计抽取方法

这种数据抽取方法主要是针对数据库中的子集合进行投影，将子集合投影到一组属性的子集合上，在投影的过程中将相同的记录进行合并和计数操作。例如，在图 9 - 1 中可是使用分组统计抽取方法，先根据还款信息表得到某借款农户已还款额的总数值，然后根据贷款信息情况表计算得到拖欠额。

③统计方法

由于抽取的数据表都是二维关系表，关系型数据库中的系统提供了五种内部聚集函数，包括 sum（）、count（）、max（）、min（）和 avg（），可以在数据采集的过程中有效地计算一些数据挖掘需要的统计量。例如，在数据抽取过程中利用统计方法，用 sum（）函数计算还款信息表，可以得到借款农户已还款数额的汇总值。

通过反复对数据进行采集处理，从原始数据库中抽取的一张能够反映业务需求和目标的二维关系表，可以称之为农户小额贷款处理数据集。这个集合共有 1000 条记录，每条记录由 15 个字段组成，包括身份证号、姓名、性别、年龄、出生日期、学历、婚姻状况、家庭人数、家庭月收入、贷款日期、贷款金额、贷款期限、月还款额、当前是否有违约债务、贷款用途以及拖欠额，如表 9 - 3 所示。

表 9－3　采集的数据字段表

属性名称	数据类型	取值范围
身份证号	字符	15 位或 18 位
姓名	字符	
性别	字符	“男”、“女”
年龄	数字	18～65 岁
出生日期	日期	1945～1992 年
学历	字符	文盲、小学、初中、高中、中专、大专、大学本科、研究生及以上
婚姻状况	字符	未婚 已婚 离异
健康状况	字符	好、一般、差
家庭人数	数值	
家庭月收入	数值	
贷款日期	日期	
贷款金额	数值	
贷款期限	数值	
月还款额	数值	
贷款用途	字符	纯农业生产 种植、养殖业、林果业 农副产品加工、农村流通业
是否有违约债务	字符	无　有
拖欠额	数值	

（2）数据预处理

通过对小额贷款数据库进行的初步的数据采集，采集后的数据可能会存在不完整的、有噪声的和不一致的，因此需要对这些数据进行数据转换，使采集的原始数据最终能够符合数据挖掘算法的要求。数据预处理过程主要包括数据清理、数据集成、数据变换和数据归约。本文根据农户小额贷款数据信息的实际特征，对采集到的原始数据进行预处理，生成挖掘数据集，使之能够进一步满足数据挖掘任务目标的需要。

① 数据清理

通过数据清理的工作可以对那些不符合建模要求、空缺值多、有误差等数据进行清洗或修正，最终得到比较整齐的、干净的数据集，可以用于数据处理和模型开发的数据样本。

采集的数据表中有很多数据项是没有值的，处理空缺值的方法主要有：忽略该条记录；人工填写空缺值；利用缺省值填补空缺值；利用均值填写空缺值；利用同类别均值填补空缺值；利用最可能的值填补空缺值。在本文所采集的数据集中，对于“婚姻状况”和“年龄”这两个字段存在空缺值，但是由于存在的空缺值比较少，因此就可以通过查询方法手工填写这些空缺值。对于不符合进行手工方式填写空缺值的数据，可以按一定的属性对数据集进行分类，然后计算有空缺值的记录所属的类中所有该属性的平均值，可以用这个计算的平均值替代空缺值。

噪声是指一个测量变量中的随机错误或偏差。采集的数据表中的字段值常常会出现错误或者偏差，可以通过范围检验、枚举清单以及一些相关的检验来完成。对于数值型和日期型的字段，可以通过范围检验数据是否在预期的范围之内。农户小额贷款数据集的数据表中，“出生日期”字段就存在部分噪声数据，有的日期出现了“1940/10/25”，这个出生日期得出的年龄是70岁，而进行小额贷款农户的年龄不能超过65岁，就可以指定他的出生日期在1945～1992年之间，找出落在指定区间之外的错误记录。枚举清单是指对照数据字段可接受值的清单检验字段的值。相关检验是将一个字段中的值与另一个字段中的值进行对比。在对比过程中，可以发现不合常理的字段值。在实际的应用过程中，这些数据处理的方法经常是结合使用。

在数据表中的字段中常常会有这样的现象：一个属性能够由另一个属性导出，则这个属性就是冗余的。此外，属性名不一致也可能导致数据集中的冗余。在农户小额贷款数据集中，“出生日期”和“年龄”具有高度的相关性，因此可以删除“出生日期”属性。此外，还可以同过计算属性的信息量处理冗余的数据。在小额贷款数据表中，“姓名”、“性别”、“健康状况”、“学历”作为农户小额贷款审评的重要指标，属性的信息量太小，因此可以将这三个属性删除。

②数据变换

属性构造。为了帮助提高精度和对高危数据结构的理解，有时需要将给定的属性构造和添加新的属性。根据“月平均收入”和“月还款额”相除，可以构造“月还款收入比”这一属性。“拖欠额”属性不能完全反应农户的欠款状况，因此可以构造“拖欠月数”属性，通过将“拖欠额”与“月还款额”相除取整得到的月数，即“拖欠月数”作为一个新的属性加入数据集中。这一新属性更加准确地反映了借款农户的信用状况，并将“拖欠月数”属性作为决策属性。

属性概化。用属性概化阈值控制技术，沿属性概念分层上卷或下钻进行概化。“婚姻状况”可以分为3类：1：未婚，2：已婚，3：离异。“学历”可以分为4类：1：研究生以上，2：本科，3：高中、中专，4：文盲。贷款用途分为3类：1：纯农业生产，2：种植、养殖业、林果业，3：农副产品加工、农村流通业。

连续属性概化为区间值。“年龄”、“家庭人数”、“月平均收入”、“月还款额”、“月还款收入比”、“贷款数额”以及“拖欠月数”这些属性都是连续型数据，对于建立决策树，用离散型数据进行处理，速度是最快的，因此需要对连续型数据进行离散化处理，根据审批人员经验和实际情况，对以下属性进行离散化：“年龄”取值划分为18～35、35～50、50～65三个区间；“家庭人数”取值划分为1～3、3～5、>5三个区间；“月还款收入比”的取值划分为<=20%、20%～50%，>50%三个区间。通过以上对小额贷款数据库的数据集进行的预处理，得出如表9-4的结果。

表9-4　农户小额贷款数据预处理结

属性名称	处理结果
身份证号	删除此属性
姓名	删除此属性
年龄	手工填写5个空缺值，经过属性概化，分别处理为：18～35、35～50、50～65三组区间值

（续表）

属性名称	处理结果
性别	“男”“女”
出生日期	该属性冗余 删除此属性
婚姻状况	手工填写 12 个空缺值，属性概化为：1：未婚，2：已婚，3：离异
学历	属性冗余，删除此属性
健康状况	该属性冗余，删除此属性
贷款日期	删除此属性
贷款金额	删除此属性
拖欠月数	构造属性，“拖欠额/月还款额”取整
月还款收入比	构造属性，“月还款额/家庭月收入”分别处理为：<=20%、20%~50%，>50%
家庭月收入	属性冗余，删除
月还款额	属性冗余，删除
贷款用途	分别处理为： 1：纯农业生产，2：种植、养殖业、林果业，3：农副产品加工、农村流通业
是否有违约债务	无　有

通过多次使用上述的数据预处理方法，从农户小额贷款数据集中构成了一个适合决策树挖掘任务的数据集，即农户小额贷款挖掘数据集。这个数据集中共有 800 条记录，每条记录由 8 个字段构成，分别为：年龄、性别、婚姻状况、家庭人数、月还款收入比、贷款用途、是否有违约债务、拖欠月数。在后面的决策树生成实现过程中，所有的工作都是基于本章所得到的挖掘数据集之上的。

9.6.5　基于决策树的信用评估模型的建立

当前处理信用风险评估采用的方法主要是采用基于分类的方法。而数据分类也是数据挖掘中的一个重要问题，也是一种有效的数据挖掘分析方法。在本文的研究中，主要是根据借款农户的一些基本信息和借款信息，将借款农户信用分为好客户和坏客户两类，这样农户小额贷款的信用风险问题就转化为分类问题。本节将采用决策树分类算法中的 C4.5 算法对农户小额贷款的信用风险建立评估模型。

在本文进行的农户小额贷款的研究中，主要是通过 C4.5 分类决策树实现定类目标变量的分类，即根据是否拖欠贷款属性来进行分类。

由于样本数据集合的数据量比较大，通过本文的数据采集和数据预处理的操作，将训练样本集中的 800 条样本分为训练样本集和测试样本集两个部分。其中训练样本集约占总样本集的 2/3（即 533 条样本），其余的 1/3（即 267 条样本）作为测试样本集合对模型进行评估验证。有了训练样本集和测试样本集之后，就可以构建农户小额贷款信用风险评估模型。

对于得到的农户小额贷款挖掘数据集来说，“拖欠月数”属性是本文研究的决策属性，通过与信贷审批人员的交流，最终得到拖欠月数超过两个月（包括两个月）以上的农户就可以将农户的属性定为“坏客户”，即不良贷款；一个月以下（包括一个月）未偿还贷款的人就可以定为“好客户”。将决策属性划分为两个类：好客户和坏客户。设有 2 个不同的类（$n=2$），类 C_1 对应于好客户，训练用样本中有类 C_1 的共有 433 条记录，类 C_2 对应于坏客户，共有 100 条记录。为计算每个属性的信息增益率，首先使用公式，计算给定样本的分类所需的期望信息：

$$\mathrm{Info}(D) = -\left(\frac{433}{533}\right)\times \log_2\left(\frac{433}{533}\right) - \left(\frac{100}{533}\right)\times \log_2\left(\frac{100}{533}\right) = 0.6962$$

然后根据公式计算每个属性的熵，本文以“农户年龄”为例，该属性有“20 岁≤农户年龄≤35 岁”和“35 岁＜农户年龄≤50 岁”以及“50 岁＜年龄≤65 岁”三种属性值，根据公式农户年龄的熵为：

$$\mathrm{Info}(\text{农户年龄}) = \frac{133}{533} * \left(-\frac{120}{133}\log_2\frac{120}{133} - \frac{13}{133}\log_2\frac{13}{133}\right) + \frac{273}{533} * \left(\frac{246}{273}\log_2\frac{246}{273} - \frac{27}{273}\log_2\frac{27}{273}\right) + \frac{127}{533} * \left(-\frac{67}{127} * \log_2\frac{67}{127} - \frac{60}{127}\log_2\frac{60}{127}\right) = 0.59064$$

农户年龄的信息增益为，根据公式（5－3）：Gain（年龄）＝Info（D）Info（农户年龄）＝0.1056

农户年龄的初始信息量，根据公式：

$$\mathrm{SplitInfo}(\text{农户年龄}) = -\frac{133}{533} * \log_2\frac{133}{533} - \frac{273}{533} * \log_2\frac{273}{533} - \frac{127}{533} * \log_2\frac{127}{533} = 1.4668$$

因此农户年龄属性的信息增益率，根据公式：

GainRatio（农户年龄）＝Gain（农户年龄）/SplitInfo（农户年龄）＝0.1056/1.4868＝0.0710

同理，可得出样本训练集中其他各属性的信息增益和信息增益率，如表 9－5 所示。

表 9－5　各属性的信息增益和信息增益率表

属性名称	信息增益	信息增益率	分支顺序
农户年龄	0.0230	0.0172	4
婚姻状况	0.0225	0.0143	5
家庭人数	0.0842	0.0550	2
月还款收入比	0.2006	0.1554	1
贷款用途	0.0724	0.0479	3
是否有其他违约债务	0.0120	0.0130	6

因为信息增益的现实含义就是从每个属性的“获得的信息量”，代表该属性对分类结果的贡献大小，其中“性别”的信息增益太小，对分类的影响也就非常小，为了避免决策树过于庞大，这个属性被删除。

根据表 9－5 所计算出的各个属性的信息增益和信息增益率，经过基于信息增益率的 C4.5 决策树算法构建决策树，并对决策树进行修剪，最终就得到如图 9－2 所示的决策树结果。

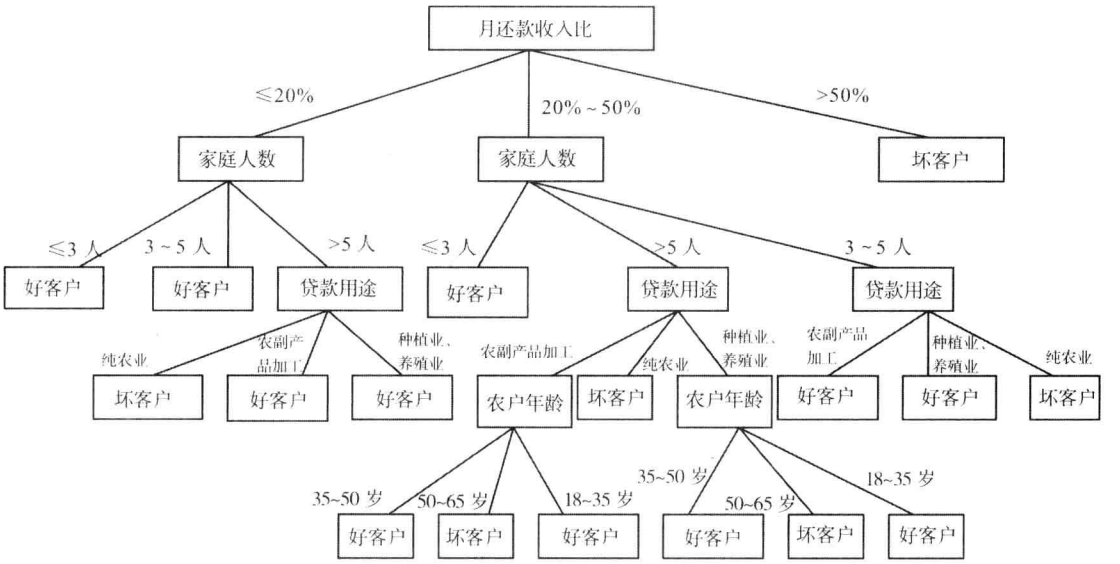


图 9-2　农户小额贷款信用评估决策树

遍历决策树，输出叶节点类属性值，用 IF-THEN 形式表达为：

（1）IF（月还款收入比 > 50%）THEN（类别 = 坏客户）

（2）IF（月还款收入比≤20% AND 家庭人数≤3）THEN（类别 = 好客户）

（3）IF（月还款收入比≤20% AND 家庭人数∈3…5）THEN（类别 = 好客户）

（4）IF（月还款收入比≤20% AND 家庭人数 > 5 AND 贷款用途 = 纯农业）THEN（类别 = 坏客户）

（5）IF（月还款收入比≤20% AND 家庭人数 > 5 AND 贷款用途 = 农副产品加工）THEN（类别 = 好客户）

（6）IF（月还款收入比≤20% AND 家庭人数 > 5 AND 贷款用途 = 种植业、养殖业）THEN（类别 = 好客户）

（7）IF（月还款收入比∈20%…50% AND 家庭人数≤3）THEN（类别 = 好客户）

（8）IF（月还款收入比∈20%…50% AND 家庭人数 > 5 AND 贷款用途 = 纯农业）THEN（类别 = 坏客户）

（9）IF（月还款收入比∈20%…50% AND 家庭人数 > 5 AND 贷款用途 = 种植业、养殖业 AND 农户年龄∈18…35）THEN（类别 = 好客户）

（10）IF（月还款收入比∈20%…50% AND 家庭人数 > 5 AND 贷款用途 = 种植业、养殖业 AND 农户年龄∈35…50）THEN（类别 = 好客户）

（11）IF（月还款收入比∈20%…50% AND 家庭人数 > 5 AND 贷款用途 = 种植业、养殖业 AND 农户年龄∈50…65）THEN（类别 = 坏客户）

（12）IF（月还款收入比∈20%…50% AND 家庭人数 > 5 AND 贷款用途 = 农副产品加工 AND 农户年龄∈18…35）THEN（类别 = 好客户）

（13）IF（月还款收入比∈20%…50% AND 家庭人数 > 5 AND 贷款用途 = 农副产品加工 AND 农户年龄∈35…50）THEN（类别 = 好客户）

（14）IF（月还款收入比∈20%…50% AND 家庭人数 > 5 AND 贷款用途 = 农副产品加工 AND 农户年龄∈50…65）THEN（类别 = 坏客户）

（15）IF（月还款收入比 ∈ 20% …50% AND 家庭人数 ∈ 3…5 AND 贷款用途 = 纯农业）THEN（类别 = 坏客户）

（16）IF（月还款收入比 ∈ 20% …50% AND 家庭人数 ∈ 3…5 AND 贷款用途 = 农副产品加工）THEN（类别 = 好客户）

（17）IF（月还款收入比 ∈ 20% …50% AND 家庭人数 ∈ 3…5 AND 贷款用途 = 种植业、养殖业）THEN（类别 = 好客户）

一共产生了 17 条分类规则。决策树的分析结果表明："月还款收入比"是决策分支最重要的影响因素，其次为"家庭人数"以及"贷款用途"，它们都是影响农户进行小额贷款信用风险程度的重要方面，应作为评估借款人信用风险的主要依据。

9.6.6 模型评估

为了检验本文所得到的基于 C4.5 决策树算法的农户小额贷款信用评估模型的效果，以及对于我国农村信用社农户小额贷款的实用性，本文把 800 条样本中的 267 个样本数据作为测试样本对模型进行评估，以验证其有效性。用测试数据集对模型进行测试：对每条客户记录，从图 9 - 2 中的决策树各节点开始，首先用农户的"月还款收入比"进行属性测试，如果农户的"月还款收入比" >50%，在决策树到达终点"坏客户"，如果农户本来分类所得结果是"坏客户"，那么模型对该农户分类正确，否则就发生分类错误；如果农户的"月还款收入比"是在 20% ~50% 之间的，到达了"家庭人数"这个节点，而"家庭人数"节点并不是叶节点，因此还要用"家庭人数"属性继续测试，最终会因为农户在该节点上的不同取值到达节点"好客户"或节点"贷款用途"。预测结果如表 9 - 6 所示：

表 9 - 6 决策树模型分类效果

	月还款收入比		家庭人数		贷款用途		全部数据	
	正确	错误	正确	错误	正确	错误	正确	错误
训练集	91.36	6.04	82.15	11.35	81.12	16.05	82.12	26.31
测试样本集	91.36	6.15	82.24	12.14	80.07	16.24	81.42	37.56

在表 9 - 6 中，节点的正确率和错误率是指被该节点正确、错误标记的记录数占节点总数据记录数的比例。其中"全部数据"一列是指在所有数据集中被正确、错误分类的记录所占的比例。为了进一步验证所建立的决策树模型的正确分类效率，再次对原始训练样本集进行 2 次取样，分别获得的训练集和测试集建立的决策树模型并对模型进行测试，得到的结果是：每次建立的模型的结构一致、各节点所属类别没有发生变化，只是在叶节点正确分类的比例上发生了一点改变。所得的结果分别如图 9 - 3、图 9 - 4 所示。其中图中黄色柱表示用训练样本集建模的情况，紫色柱表示用测试样本集进行测试的结果。

通过本节所采用的测试样本集进行测试的结果可以看出，模型的表现比较稳定：正确分类的比例和建模使用的训练样本集的结果以及 3 次的实验结果都没有明显的差别。

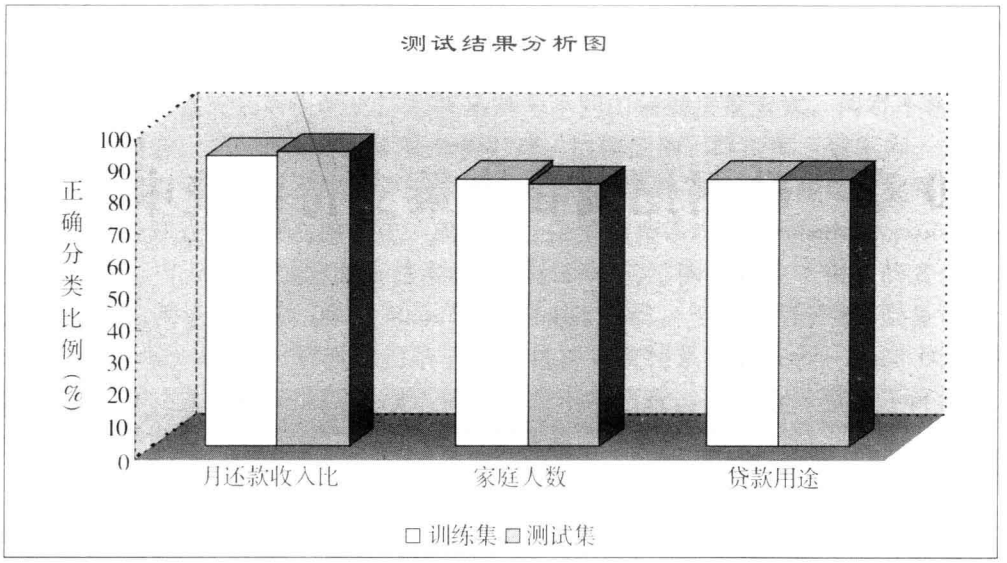


图 9－3　第 2 次试验结果

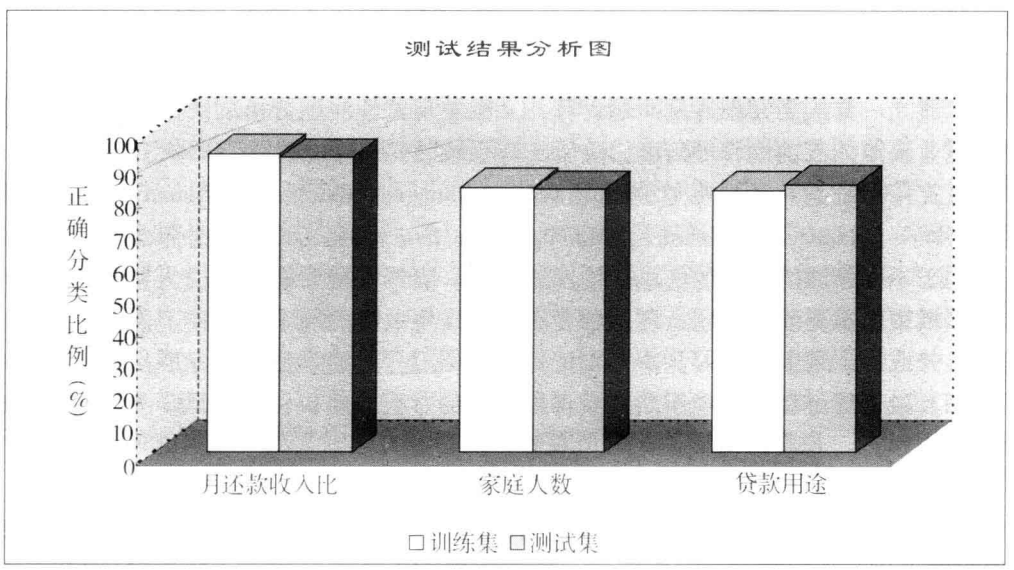


图 9－4　第 3 次试验结果

至此，我们获得了通过农户填写的申请表以及农户以往贷款的信息与该农户好坏的评定之间的模型，如图 9－2 所示，并且通过这个决策树模型得到了 17 条分类规则，使用这些规则表示决策树模型，使得该模型能够更好地被业务人员理解。通过建立的决策树模型以及所得出的分类规则，可以对农户的一些信息（包括：月还款收入比、家庭人数、贷款用途等）与农户是否是一个“好客户”之间的关系进行分析。“月还款收入比”是影响农户是否能够按期还款的一个重要因素，如果月还款收入比率大于 50%，就极有可能是一个“坏客户”；“家庭人数”也是一个重要的影响因素，其中家庭人数小于 3 人的农户家庭，农户拖欠贷款的可能性就很小，成为“好客户”的可能性就越大；对于“贷款用途”这个属性中对于贷款用来进行纯农业生产来说，不能按时还款的可能性比较大。

第 10 章　数据挖掘在决策支持系统中的应用

10.1　决策支持系统

由于现代企业的生产经营活动所涉及的影响因素非常多，决策行为往往带有很大的不确定性，在目前激烈竞争的市场环境下要做出正确的决策并不容易。决策支持正是为了帮助管理者在变化如此之快、如此不确定的环境中，做出正确的决策，使得管理者能够有效地工作。大量的实践表明，在决策过程中，运用管理学中的理论和先进管理思想，通过计算机技术建立一套既能完成定量问题计算，又能实现定性问题分析的决策支持系统，能有效提高企业决策的准确性和可信度，决策支持系统就是一个非常强有力的工具。

决策支持系统是在传统的管理信息系统（Managementhiformationsystem，Mls）和管理科学/运筹学（Management Seience，MS/Operation Research，OR）理论基础上发展起来的一门适用于不同领域的全新的信息系统发展分支。决策支持系统已经成为系统工程与计算机应用领域中的重要研究课题，各国学者在决策支持系统的理论与应用上进行了卓有成效的工作，并取得了很大的进展。决策支持系统是一门开放的学科，其发展历程是与其他学科不断相互融合的过程，也是不断拓展自身内涵的过程。简要说来，DSS 大致经历了这样几个发展历程：

20 世纪 70 年代初，决策支持系统的概念由美国 M. S. Sott. Morton 教授首先提出。20 世纪 80 年代，Sprague 提出了决策支持系统三部件结构（对话部件、数据部件、模型部件），明确了决策支持系统的基本组成，极大地推动了决策支持系统的发展。当时的 DSS 是由模型库系统、数据库系统与人机交互系统组合而成的，通过模型的组合对大量数据进行处理，从而得到辅助决策信息，达到支持决策的效果。1981 年，Bonczek 等人提出了决策支持系统（DSS）的三系统结构形式，即它由语言系统（LS）、知识系统（KS）和问题处理系统（PPS）三个部分组成。由于该系统中包含“知识系统”，从而使决策支持系统包含人工智能的成分。这使很多 DSS 学者采用了这种决策支持系统的结构形式。20 世纪 80 年代末 90 年代初，决策支持系统开始与专家系统（Expert System，ES）相结合，形成智能决策支持系统（Intelligent Decision Support System，IDSS）。智能决策支持系统充分发挥了专家系统以知识推理形式解决定性分析问题的特点，又发挥了决策支持系统以模型计算为核心的解决定量分析问题的特点，充分做到了定性分析和定量分析的有机结合，使得解决问题的能力和范围得到了一个大的发展。20 世纪 90 年代出现了数据仓库、联机分析处理和数据挖掘新技术，DW + OLAP + DM 逐渐形成新决策支持系统的概念。为此，将智

能决策支持系统称为传统决策支持系统。新决策支持系统的特点是从数据中获取辅助决策信息和知识，不同于传统决策支持系统用模型和知识辅助决策。

传统决策支持系统和新决策支持系统是两种不同的辅助决策方式，两者不相互代替，更应该是互相结合。把数据仓库、联机分析处理、数据挖掘、模型库、数据库、知识库结合起来形成的决策支持系统，即将传统决策支持系统和新决策支持系统结合起来的决策支持系统是更高级形式的决策支持系统，成为综合决策支持系统（Synthetic Decision Support System，SDSS）。近年来，随着网络技术的发展，DSS 的结构也发生了很大的变化，总的看来，经历了从基于主机的 DSS（Mainframe - based DSS）至基于客户机/服务器的 DSS（Client/Server - based DSS）再到目前基于 Web 的决策支持系统（Web - based DSS）的变化过程。被称作“第三次互联网浪潮”的网格技术的出现，又为 DSS 带来了许多先进的思想理念和创新技术，必将给 DSS 的发展带来新的发展契机。

DSS 的概念最早是在 20 世纪 70 年代初期，由 Scott Morton 教授在其著作《管理决策系统》一文中首先提出。但是，由于人们对 DSS 的认识不完全相同，所以至今还没有一致公认的 DSS 定义。早期 DSS 的定义表明 DSS 是在处理半结构化的问题中支持决策人、扩展决策人的能力，但不代替其判断。Little 将 DSS 定义为支持管理者决策、数据处理和判断、应用模型的一组过程。他认为，为了取得成功，系统必须是简单的、鲁棒的（Robust）、容易控制的、对重要的问题是完全的以及容易与其通信。隐含在该定义中的假设是基于计算机的系统，该系统为用户提供服务，以扩展用户求解问题的能力。

Bonczek 等将 DSS 定义为由三个相互联系的部件组成的基于计算机的系统，即语言系统，它提供用户与 DSS 其他部件相互通信的机制；知识系统存储在 DSS 中的有关问题领域的知识；问题处理系统，它连接其他两个部件，并包含决策所需要的一个或多个一般问题处理功能。该定义提出的概念对于理解 DSS 和 ES 的结构以及两种系统的关系是重要的。

在这里，我们给出一个 DSS 的定义，即决策支持系统是一个交互式的、灵活的和自适应的基于计算机的系统，它综合应用数据、信息、知识和模型，并结合决策人的判断，支持决策过程的各阶段，支持决策者进行半结构化和非结构化决策问题的分析。

随着计算机技术的不断发展，人们能利用计算机来完成很多的任务，用计算机辅助管理者决策，是科学家总结了信息技术在管理活动中应用的成功和失败两方面的经验之后，提出的一个新的应用模式。决策支持系统的目的是为决策者提供决策支持，而决策支持是通过决策过程来实现的，决策过程是一个提出问题、分析问题、求解问题和评价问题的过程。对于现代的新型决策支持系统，大都采用数据仓库与数据挖掘技术，新的决策支持系统的体系结构如图 10 - 1 所示。

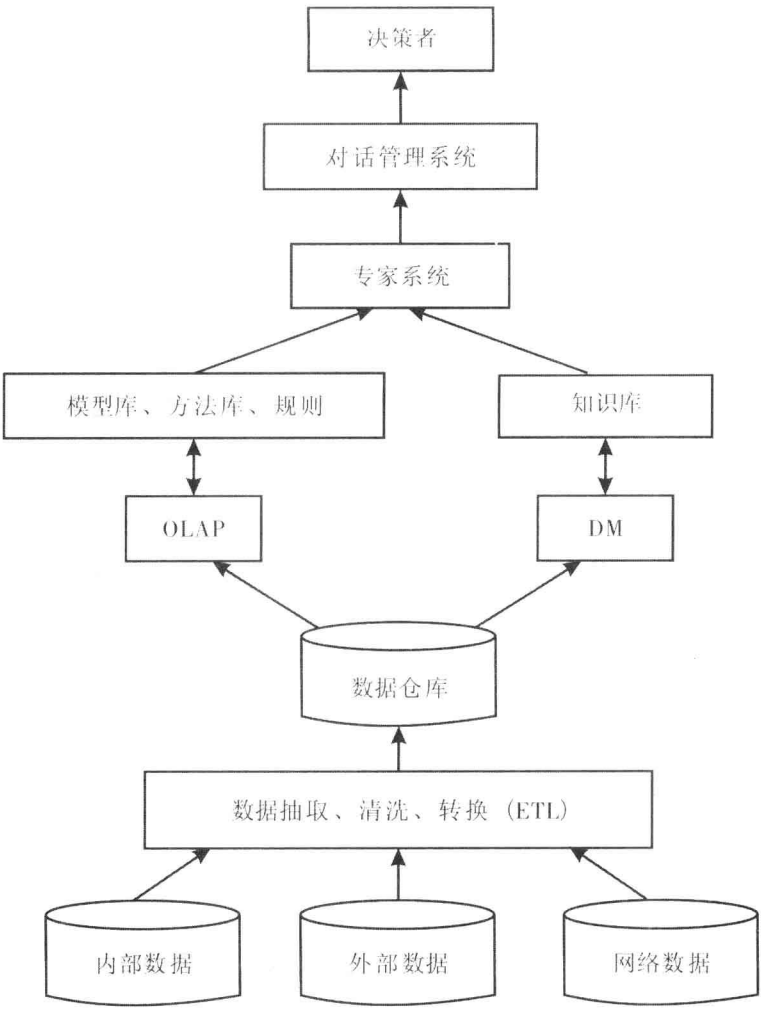


图 10-1　决策支持系统结构图

10.2　决策支持系统的主要特征及功能

（1）通过前面的文献综述，可总结出决策支持系统（DSS）的主要特征：

①由于半结构化决策问题和非结构化决策问题都含有一些定性因素，这就造成了在用公式或者规则来解决这类问题时会遇到很大麻烦。尽管 DSS 可以在分析工作中辅助管理人员解决系统化问题，但在具体做决策时还是需要决策者进行一定的分析和判断，这部分的有效性就取决于决策者自身的知识与经验。

②DSS 并不能代替管理人员进行决策判断，而只是一种辅助和支持工具。所以，我们不应该期望计算机会直接给出最终答案，也不能将那种事先编定好的分析逻辑灌输给决策者。

③DSS 对决策者提供的决策支持功能，是通过它的人机接口来实现的。DSS 在辅助决策者进行决策时，需要决策者不但要主动利用系统提供的各项功能，而且还要决策者自己具有一定的实际经验和判断力，然后再通过系统提供的支持功能进行反复地探索和试验，

最终根据决策者的“管理判断”确定一个最佳方案。

④DSS 的目的并不是取代那些能够提高管理效率的数据处理系统或者管理信息系统，而是改进决策制定的效能以及在决策者的决策过程中提供辅助性工作。

（2）决策支持系统（DSS）的功能总结如下：

①在辅助进行某一决策时，决策支持系统整理系统中诸如库存、财务状况以及生产能力等相关的各种数据，并及时提供给决策者；

②在辅助进行某一决策时，决策支持系统能够主动收集和存储诸如新技术、原材料价格波动以及市场需求变化等本系统之外的各种相关数据，并及时提供给决策者；

③决策支持系统能够及时收集诸如生产计划完成情况、客户反映或者产品销售状况等各种反馈信息，并将它们提供给决策者；

④在决策支持系统中，存储了与决策相关的各种数学模型和模拟模型，例如总资产报酬模型、择优定价模型等；

⑤决策支持系统中存储了常用的数学方法和运筹学方法，在需要时可以及时提供给决策者进行相关的统计检验、线性规划以及回归分析以及其他的数理统计分析；

⑥系统的使用者可以直接修改系统中存储的各种模型、方法和数据等，或者向系统中添加相关信息，能够很容易地对系统进行管理；

⑦系统的使用者还能够灵活地使用系统中存储的各种模型和方法对数据进行汇总、加工等，还能够进行分析和预测，以便得到进行决策所需要的各种综合信息和预测信息；

⑧系统不但提供有方便的人机对话接口，还能提供图形输出功能，不但能够为使用者提供随机查询所需要的数据，而且系统还能回应诸如“what…if…（如果…则…）”之类存在逻辑性的指令；

⑨系统的运算速度和加工时间是使用者能够承受的。

10.3 智能决策支持系统

10.3.1 智能决策系统概述

智能决策支持系统（Intelligent Decision Support System，IDSS）是以传统决策支持系统为基础发展起来的一种基于智能化的和知识化的决策支持系统。智能决策支持系统的核心思想是将人工智能以及决策支持系统相结合，并应用专家系统技术，使决策支持系统能够更充分地应用人类的专家的知识通过逻辑推理来解决复杂的决策问题的辅助决策系统。

智能决策支持系统是由数据仓库、模型库、OLAP 技术、数据挖掘以及交互接口集成在一起的。其中，数据仓库能够实现对决策主题的存储以及综合；OLAP 可以对多维数据进行分析；数据挖掘就是从大量的数据中提取或者挖掘知识；模型库能够实现多个广义模型的组合辅助决策；专家系统则利用知识推理实现定性分析。它们之间相互补充、相互依赖、发挥各自的辅助决策优势，实现更有效的辅助决策。可以根据智能决策支持系统的智能实现将 IDSS 分为 3 类。

第一类：智能决策支持系统中的智能是由 AI 实现的。它包括利用 ES 实现智能的 IDSS，利用机器学习实现智能的 IDSS 以及利用 AI 实现的 IDSS。在这其中，目前 AGENT

的研究已经成为这一类中的热点。

第二类：智能决策支持系统中的智能是利用数据挖掘中的一些工具实现的。这些数据挖掘工具包括数据仓库、OLAP 等。通过使用这些数据挖掘工具，可以使用户有效地处理复杂数据，并基于得到的结果做出正确的判断。

第三类：智能决策支持系统中的智能是利用面向对象方法、逻辑方法或者关系来表示 IDSS 的各个部件，使得 IDSS 的各个部件能够相互协调并且相互统一，整体实现 IDSS 的智能。

10.3.2 智能决策支持系统的特点

（1）智能决策支持系统具有一定的自主学习的能力，它允许决策者修改以及扩充知识库当中的知识，从而使得其解决问题的能力得到进一步提高。

（2）智能决策支持系统具有推理机构，其能模拟决策者的思维过程，根据决策者的不同需求，通过人机会话，应用有关的知识引导决策者以选择合适的决策模型。

（3）智能决策支持系统具有智能的模型管理功能。其将模型作为一种知识结构进行管理，简化了各个子系统之间的接口。

（4）智能决策支持系统建立了更为通用的决策支持系统的结构，用于扩大系统的服务领域，也使系统能对环境的变化以及决策方式的变化具有一定的适应能力。

10.3.3 智能决策支持系统的基本体系结构

智能决策支持系统（IDSS）的结构是在传统的决策支持系统（DSS）三库结构模型（模型库，数据库，方法库）上通过增加知识库和推理机，在人机对话接口上增加语言处理系统而形成的四库系统结构。人机会话接口是决策支持系统（DSS）与用户以及知识工程师进行通信的界面，它的主要职责是从工程师那获取知识，响应用户的请求，提供决策信息给决策者。语言处理系统是沟通用户和系统的纽带和桥梁，所有用户提出的问题均要通过语言处理系统来描述并且响应。问题处理系统是智能决策支持系统（IDSS）中不可缺少的部分，其主要完成系统的动态问题的求解过程，即接受用户提出的各种各样的问题，利用知识库中的专家知识，得出求解的过程。推理机则是在专家系统中将一些部件在计算机中的实现，这些部件是基于知识推理的部件，主要包括推理以及控制这两个方面。知识库用于存取以及管理所获取的专家知识和经验，以供推理机利用，有知识存储、编排、检索、修改、增删和扩充等功能。对话子系统与模型库子系统的交互作用，可使用户控制对模型的操作、处置以及使用；它与数据库子系统交互作用，以便提供各种模型所需的数据，实现模型输入、输出以及中间结果存取自动化；它与方法库子系统交互作用，实行灵敏度分析、目标搜索以及仿真运行自动化等。模型库子系统的主要作用则是通过人机交互语言使得决策者能够方便利用模型库中各种各样的模型支持决策，其引导决策者应用建模语言以及自己所熟悉的专业语言建立模型、修改模型和运行模型。典型的智能决策系统的体系结构如图 10－2 所示。

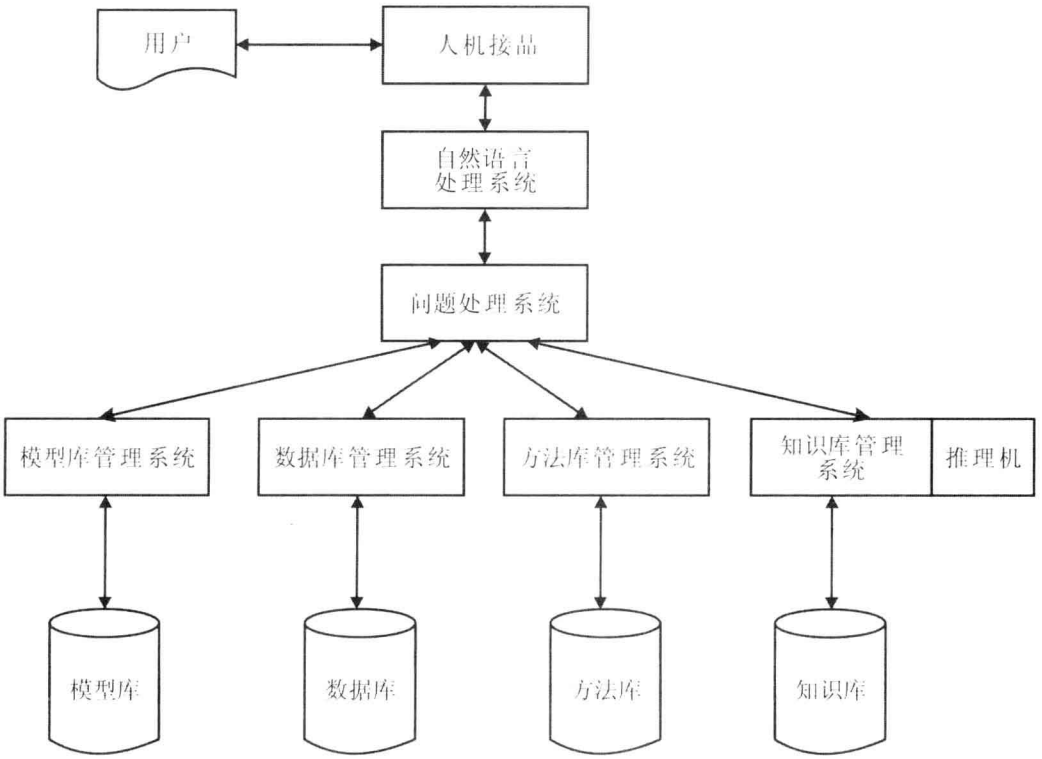


图 10-2　智能决策系统的体系结构

10.4　基于数据挖掘的房地产决策支持系统的开发

房地产信息系统在近几年已逐步完善，随着互联网的快速发展，越来越多的人通过互联网来了解房源市场，在原有房地产信息系统的基础上开发基于 Web 的决策支持系统的需求越来越强烈。本节研究在中国房产信息集团（简称 CRIC，China Real Estate Information Company）信息系统的基础上，设计基于数据仓库和数据挖掘的房地产决策支持系统。采用多维数据仓库技术，运用 OLAP 联机分析处理和数据挖掘 DM（Data Mining）等比较先进的技术来进行系统的开发，并采用原型法开发模式。整个系统的实现过程坚持“统一规划、分步实施、步步见效、逐步完善”的原则。

10.4.1　系统需求分析

（1）业务需求

房地产决策支持系统的研究目标为：通过实现房地产决策支持系统，将中国房产信息集团积累的历史数据与现代互联网网络访问用户行为的结合，积极探索人们对房地产概念的认识以及各类层次用户对房源的不同需求，挖掘楼市热点，给内部各级管理人员，以及中介机构提供决策支持。

具体的研究目标如下：

①在数据仓库技术、数据挖掘技术、联机分析处理技术和决策支持系统技术的基础

上，明确系统的设计原则和方法，并设计房地产决策系统的框架。

②以中国房产信息集团的信息系统作为研究对象，在系统框架的基础上设计完善系统的主要功能，包括系统的体系设计与功能设计。并完成数据仓库的设计，包括数据仓库的建模，记录系统的定义以及数据的生成、抽取、清洗、转换和装载。

③在深入理解系统功能的基础上，建立一个集成的、共享的房地产决策支持系统平台，为中国房产信息集团各相关部门提供决策服务。

该平台应是面向主题的，应具有足够的灵活性，能适应不断变化的政策和用户需求，能适应各种不同的用户群体的要求。该平台要能涵盖用户信息以及网络访问行为的数据和信息，并具有通用性、可扩展性、技术开放性和兼容性等现代决策支持系统的优势。

（2）功能需求

房地产决策支持系统的功能需求有以下三类：

①数据分析，通过数据分解和汇总对已经具有的数据进行计算处理。如注册用户的地区、年龄、性别的分布；经纪人（包括机构）套餐使用费用汇总；网站的总流量，独立访问人数，平均访问时长，平均访问深度等。

②决策建议，根据网站访问的情况，页面访问情况，判定热点页面，着陆页和离开页的日趋势图，提出页面开发与布局建议和调整推广策略的建议，根据用户套餐订购情况，设置某些套餐使用不达标的预警，提出开发新套餐建议。

③预测，根据现有的数据，研究其变化趋势，对未来作简单预测分析。例如：预测网络运行未来的发展趋势，楼市热点的变化趋势。

根据需求的不同类别，我们在实际应用中把分析类型分为三类：主题分析、专题分析、预测分析。

主题分析是面向分析主题按一定周期在格式相对固定的分析模板基础上进行的数据分析。其模式类似于固定格式的统计报表，但是超越统计报表。主题分析工作的目的是通过固定模式的分析跟踪主题业务发展动态，发现变化发展的趋势，判断其是否合理，从而发现问题并寻求引起问题的因素，最终提出解决问题的措施。主题分析对于分析人员就某项专业分析经验的积累、敏锐眼光的培养有一定意义。数据仓库通过各个主题下的定制报表和 OLAP 多维报表查询支撑主题分析，反过来，主题分析常常是作为定制报表和 OLAP 多维报表查询的需求在数据仓库中固化并得到实现。

专题分析是根据实际业务过程中出现的热点问题以及主题分析发现的异常情况而确定，需要进一步深入进行的分析。专题分析没有固定的模板，因此每执行一次专题分析需要向数据仓库提交一次数据需求，包括多维度的数据表需求，而数据仓库则是通过临时创建的多维报表以及符合条件的数据来支撑专题分析工作的进行。

预测分析一般放在模型管理中，作为固化模型进行维护和分析。

以下是房地产决策支持系统主题分析。

①用户情况分析：用户情况分析可以细分成如下两种。

a. 注册经纪人分析

注册经纪人分析主要是了解经纪人所属的房地产经纪公司，是否诚信通认证用户，使用的套餐，已经消费的总额度，经纪人的所属地区。可分析得出“某经纪公司对网络系统的贡献大小”，用于辅助决策。有助于市场运营部门开展业务。

b. 普通注册用户分析

普通用户分析主要是了解用户对网站的熟悉程度，网站各频道的使用率，可分析得出哪些是热点频道，哪些频道需要进一步改造，提供分析结果给策划设计部门，以提高网站对用户的黏着度。

②网站运行情况分析

对于网站运营情况的考核，可以参照 Alexa 的标准来定义一些参数：

如 PV（PageView 的缩写）表示本站网页被浏览的总页面数；UIP（Unique IP，即独立的 IP 地址）表示访问到本网站来的外部的不同的 IP 地址数；PageView/User 表示用户平均访问深度，表示每个独立的用户进入本网站后平均访问过的页面数；Bounce 表示跳出率，即进入本网站，但没有进一步访问到任何别的页面的人数与总进入人数的比例，亦即访问深度为 1 的用户与总访问用户的占比。（用户数的判别可以通过独立 Session 会话数或独立的 Cookie 数来确定）另外考虑到用户从哪个页面入口进入本站和从哪个页面出口退出本站，以及用户使用的某种浏览器或是操作系统的考核指标，可以定义更多参数。比如，着陆页：进入本网站访问的第一个页面。离开页：从哪个页面退出本站。热点页：用户点击量最多的页面，可以以页面的 PV 数来衡量。

（3）运营部门 KPI 指数需求

对于网站运营情况，数据部门要提供给运营部门的数据指标有：

人数考核指标：

①每日平均在线人数，指 15 分钟为时间段的在线人数。

②月注册人数，是指一月内的注册人数，分成两类，一类是注册的普通用户数，另一类是注册的经纪人（中介机构）数。

③每日平均活跃用户数，是指当日登录的用户数（包括当日新注册）频道考核指标：频道主要分为二手房、租房、写字楼，小区考核的指标包括频道访问人数、平均访问时长、平均访问深度，KPI 指数要求平均日访问人数不低于 1000 人，平均访问时长在 5 分钟，平均访问深度为 5 个页面。未达到指标发出警告，以便运营部门调整运营策略。

10.4.2 房地产决策支持系统总体设计

根据房地产决策支持系统的需求，房地产决策支持系统总体结构由源数据层、数据转换层和分析展示层三个层次构成，如图 10 - 3 所示。

（1）源数据层

包括目前正在运行的应用，如各种基于关系数据库的事务性应用（包括每日新增经纪人统计，经纪人在网上发布房源数，房源刷新次数，房源被热推次数等）、财务管理方面（包括经纪人套餐购买统计，每月收益统计，业务员推广收益等）、办公自动化管理应用等，实现日常的业务处理，这些系统所处理的数据是数据仓库的数据源。线上数据是指网站运行产生的日志数据；线下数据是指业务系统产生的数据的统称。数据源的类型包括关系数据库、办公数据、平面文件等。

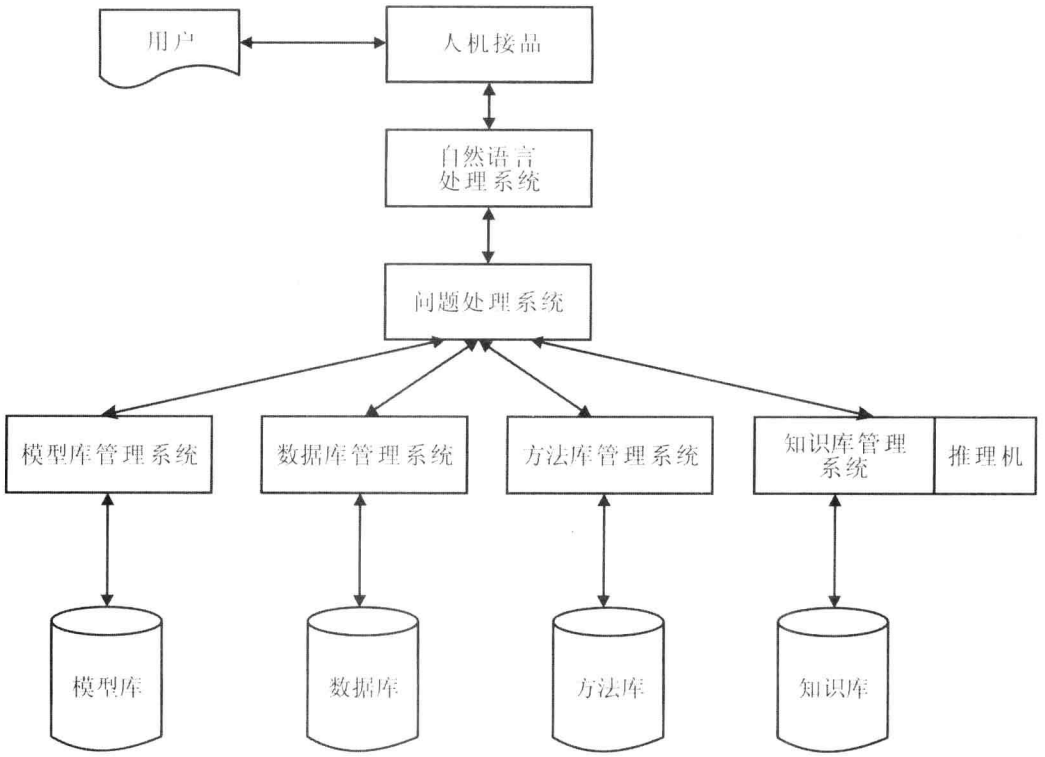


图 10-3　房地产决策支持系统总体结构图

（2）数据转换层（ETL）

实现数据抽取、转换和加载过程。将原始日志通过清洗，分析转换成数据库中的表数据，包括对线上日志及导航路径数据表的加载，以及实体维度表的加载。

数据仓库由面向不同主题的数据集市构成。在本房地产决策支持系统中，数据仓库包括两个方面的主题，一是与 Web 日志相关的网络用户产生的点击历史，一是业务方面的基于经纪人的客户数据。

（3）分析展示层

通过运行预定义存储过程，根据业务需求，整合成能提供决策支持的数据并加载入 Web 数据仓库中，数据仓库中保存着与决策相关的各类历史数据，以满足各级决策需求。分析者利用数据仓库中的数据进行时间序列分析，用户行为习惯分析，根据历史记录来预测销售业绩。如何判断楼市热点（房源热点），可以根据 Web 访问量的集中度来将楼市热点分类；也可对网络用户进行分类，根据网络用户访问页面的习惯来判断属于哪一类推广对象。

10.4.3　房地产决策支持系统 ETL 设计

ETL 设计的目标是以最小代价（包括对日常操作的影响和对技能的要求）将针对日常业务操作的数据转化为针对数据仓库而存储的决策支持型数据。本课题研究与开发过程中采用的是 SQL sever Intergration services 工具。在房地产决策支持系统中，ETL 包括两方面数据的 ETL 转换。一方面是线上日志数据，一方面是线下业务数据。

（1）日志预处理——线上数据

Web 日志中蕴涵着大量的信息，这些信息的存储、跟踪和挖掘最早用于系统故障的诊断。如今，Web 日志文件已广泛应用于入侵检测，缓存性能优化，频繁访问模式分析和网页访问预报等。本节中，房地产互联网信息平台的数据仓库的架构正是基于 Web 日志的分析的。由于日志数据是一种半结构化（Semi - structured）的数据类型，因此在进行加载之前需要对其进行清洗和整理。目前 Web 负载中通常存在大量的爬虫请求，我们将爬虫请求分为两类：

①可从字面识别的爬虫

对于第一类爬虫，我们通常不会在服务中禁止其请求。这是因为第一类爬虫通常是商业搜索引擎所发出的爬虫，网站出于搜索引擎收录的需要一般不会将其屏蔽。然而在分析普通网民的访问行为时，该类数据请求的存在会“污染”统计分析结果。因此，在洗过程中，我们应根据其字面关键字过滤掉该类爬虫。

②非可识别的爬虫信息

第二类爬虫通常是商业竞争对手所发出的爬虫，其目的在于大规模的获取有价值的商业信息。然而由于这类爬虫不含有字面信息从而达到伪装的目的，因此我们需要针对其特征制定相应规则来进行过滤。

除了爬虫请求需要被过滤以外，一些提取数据的请求是不通过与用户的交互而被自动触发，例如异步加载的 JavaScript 和 XML（Ajax）由于这些请求不能体现用户行为，因此该类请求也需要在预处理过程中被过滤掉。此外，由于 IP 不能用来甄别唯一用户，因此我们利用 IP 结合 Cookie 的识别机制对用户身份进行识别。Cookie 存在于客户端，是服务器用来识别客户端的一种方法。然后 Cookie 有可能被客户端关闭，因此该方法仍存在一些局限性。通常的做法是设置一个阈值，例如，30 分钟，如果对同一个用户或者同一个 IP 地址，一条日志记录同下一条记录之间的时间间隔不大于该阈值，则认为这两条记录属于同一个会话。

将日志分析的预处理分为三个阶段：清洗解析阶段、导航路径提取阶段、加载阶段。在清洗解析阶段，日志文件首先根据预定义的规则被收集并过滤，然后根据网站页面内容和对应的 URL 地址进行解析。之后，我们提取用户的导航路径并将其转换为事务集，最后一个阶段将清洗解析后的日志和用户导航路径事务集加载至数据仓库。如图 10 - 4 所示，ft_ log_ global 为解析后的用户点击全集，ft_ log_ global_ path 为解析后的用户导航路径，亦即为用户进入网站的按顺序的点击行为集。

（2）业务逻辑数据整合——线下数据

房地产信息平台线下数据一般为与业务逻辑相关联实体的属性信息，例如所登记注册的房源信息，所属经纪人或个人的身份信息，土地，开发商以及楼盘所在位置的交通信息等。在建模过程中，线下数据通过被用来建立与用户行为对象相关联的维度。由于这些业务逻辑数据在网站开发和运营过程中存在于网站后台数据库关系表中，其结构往往是按照开发逻辑而设计的，因此并没有考虑到将来数据挖掘和分析的问题。解决此问题行之有效的办法是将线下数据重新整合，形成可直接聚合的事实数据和维度数据。在本系统研究与开发过程中，我们设计了另一个单独的线下数据 ETL 过程。通过这个 ETL 过程，散落在数据库冗杂的数据表中的信息被重新按照实体和事实之间的关系组合了起来，并且被推送到新的数据表中，作为线下数据的事实表和维度表。

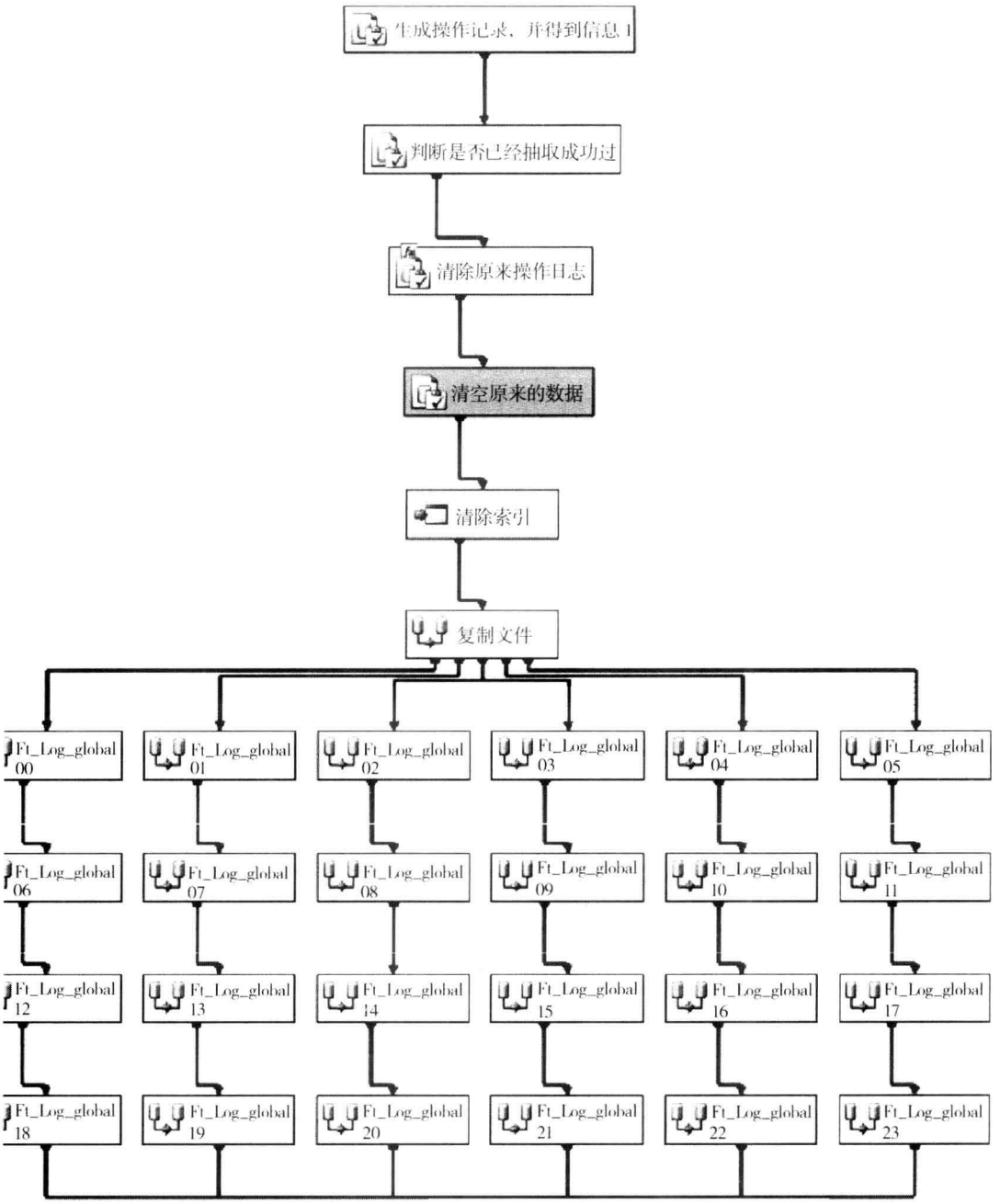


图 10－4　日志文本转换成 SQL Server 数据库文件

如图 10－5 所示是线下数据 ETL 的一个示例，业务数据中经纪人信息表分别在不同的服务器上的数据表中，有着相同的格式，在数据仓库中要分析这部分数据。为了方便，将位于不同服务器上的数据表抽回到专门的数据服务器上。

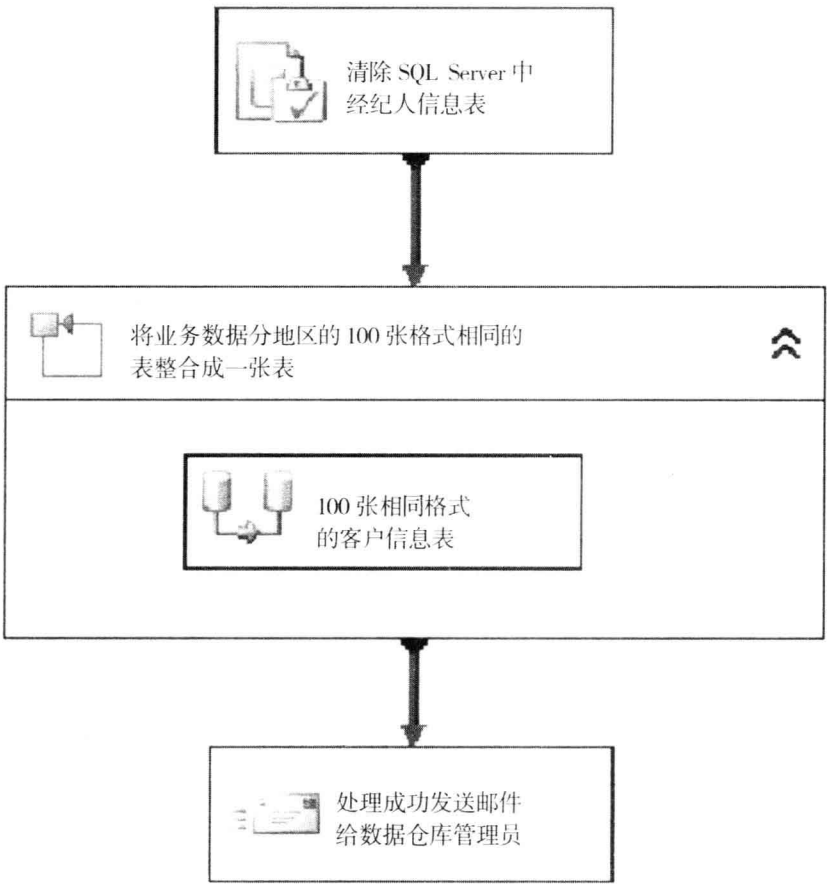


图 10 - 5　线下数据 ETL 示例

10.4.4　房地产决策支持系统数据仓库设计

星形模型是一种由一点向外辐射的建模范例，中间有一对一对象沿半径向外连接到多个对象。星形模型中心的对象称为“事实表”，与之相连的对象称为“维度表”。对事实表的查询就是获取指向维度表的指针表，当对事实表的查询与对维度表的查询结合在一起时，就可以检索大量的信息。通过联合，维度表可以对查找标准细剖和聚集。CRIC 决策支持系统中的数据仓库，其采用了星形结构，如图 10 - 6 和图 10 - 7 所示。

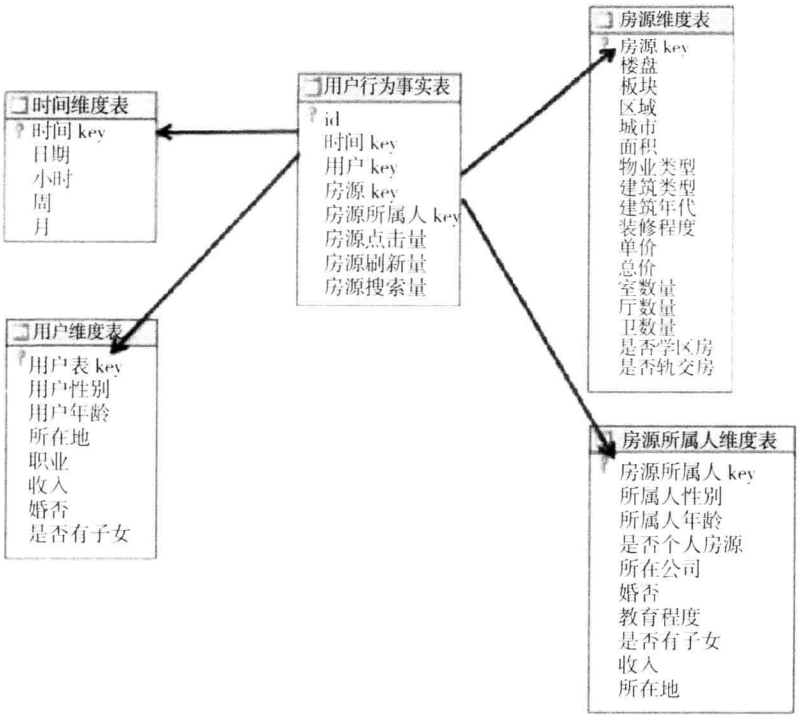


图 10－6　用户行为事实数据与实体维度

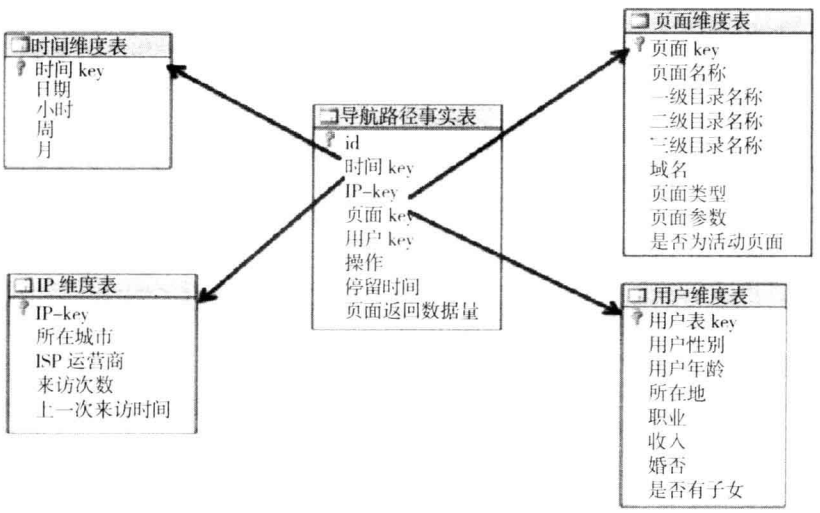


图 10－7　导航路径事实数据与实体维度

10.4.5　决策支持数据挖掘设计

粒度：粒度问题是设计数据仓库的一个最重要方面。粒度是指数据仓库的数据单位中保存数据的细化或综合程度的级别。细化程度越高，粒度级就越小；相反，细化程度越低，粒度级就越大。例如：考虑到网络推广过程中，可能存在欺诈行为，在实际设计过程中，以小时为段仍不能准确把握网络的异常情况，则将时间粒度进一步细化到一刻钟，以15 分钟在线活跃人数来进行判断，监测网络用户的异常情况。

通过向导的方式，用户可以定义分析因素的汇总行，例如对于各地区各年度的销售情

况，可以生成地区与年度的合计行，也可以生成地区或者年度的合计行。而 drill down 则相反，它从汇总数据深入细节数据进行观察或增加新维。例如，用户分析“各地区、城市的销售情况”时，可以对某一个城市的销售额细分为各个年度的销售额，对某一年度的销售额，可以继续细分为各个季度的销售额。通过钻取的功能，使用户对数据能更深入了解，更容易发现问题，做出正确的决策。钻取允许驾驭一个报表内的不同层次的信息。

在进行数据挖掘层系设计时，将各个维度表按照业务需求来划分层次关系，这样在进行钻取时，就能非常容易地获取到钻取的各级汇总或分类报表。

客户层系：如图 10 - 8 所示。包括客户的年龄、性别、区域、收入情况及订单 Order 等属性。可以挖掘出各个属性的分布规律及变化趋势。

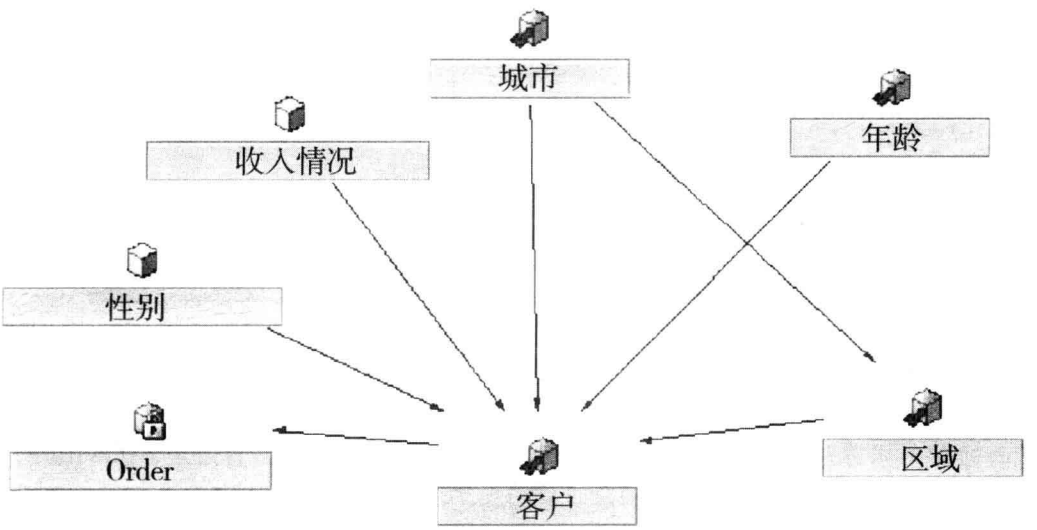


图 10 - 8　客户维层系图

时间层系：如图 10 - 9 所示。数据挖掘过程中建立对历史数据的时间序列分析是必须的，建立时间层次，就是便于对日趋势、周趋势、月趋势以及年趋势的分析，根据时间线上的趋势来把握网络运行状况。在具体的在线人数判断中，以 15 分钟在线人数来作为当前在线人数，所以在时间层系中细化到每刻钟。

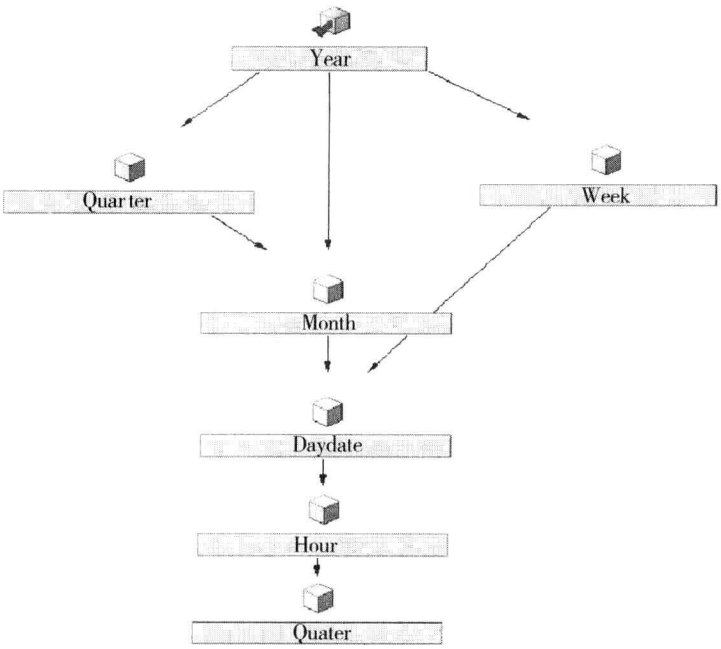


图 10 - 9　时间维层系图

地区层系：如图 10 - 10 所示。便于对房源及客户的地理维分布的分析。

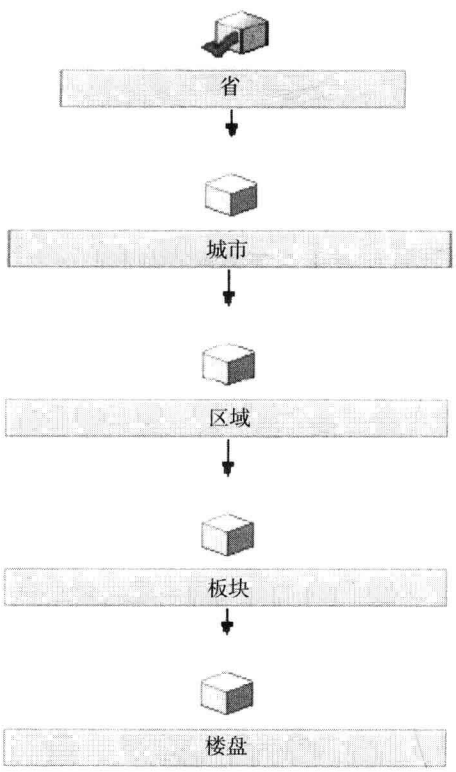


图 10 - 10　地区维层系图

网站 URL 结构层系：如图 10 - 11 所示。可在域名以及各级目录上进行挖掘，分析出页面对网站的指数贡献，利于策划部门对网站页面目录调整的设计。同时一级目录又表示了“城市”，这样就很容易对各省市的网络用户行为进行分割，从而分析出城市指占比，为运营部门的业务拓展提供决策支持。

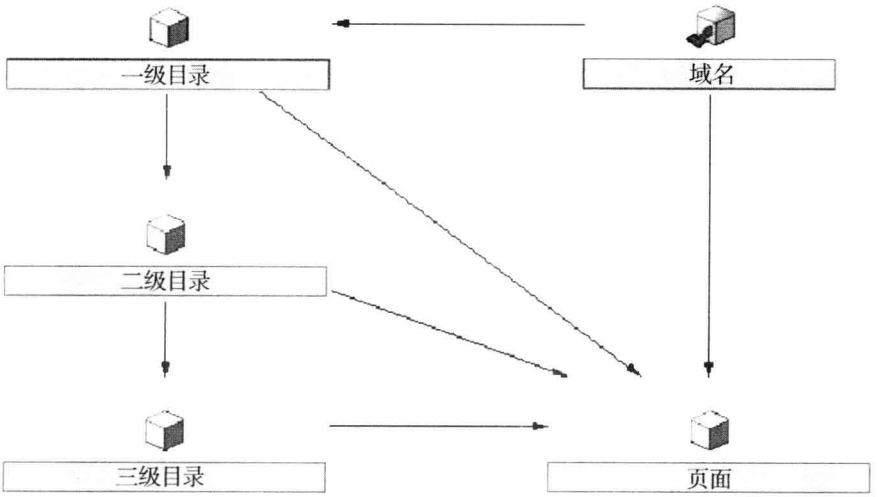


图 10 - 11　URL 层系图

10.4.6　决策支持报表系统 OLAP 设计

利用 MSTR 来设计与开发 OLAP 前端展现报表。MSTR 元数据库是数据从数据仓库转移到 MicroStrategy 应用程序进行使用的相关信息。它保存了 MicroStrategy 的对象定义、数据仓库的相关信息和 MicroStrategy 对象与数据仓库的结构、内容映射的关系。MicroStrategy 应用程序使用元数据库将用户的请求翻译成 SQL 查询语言，并把那些执行的查询结果转化回 MicroStrategy 的对象，例如报表和文档对象。

10.4.7　房地产决策支持系统的实现

（1）系统开发运行环境

硬件环境：网站的拓扑结构采用的是 B/S 架构，北京、上海、深圳三地均有分布式数据服务器。因为网络用户每日访问产生的数据量均在数百 M 或 G 级，在硬件配置上，推荐用 IBM 两台双机容错加磁盘阵列柜的专用数据库服务器和两台均衡负载的专用业务服务器，来运行此房地产决策支持系统。对于使用决策支持系统的用户提供登录 ID 和密码，普通微机即可。

软件环境：应用信息系统开发应用的环境为 Linux，MySQL，Apache；数据仓库与数据挖掘系统开发与运行环境为：Windows 7/XP，SQL Server 2008，Python，MS SqlServer 2008 Business Intelligence；报表系统主要采用 Microstrategy。

系统部署拓扑结构如图 10 - 12 所示。

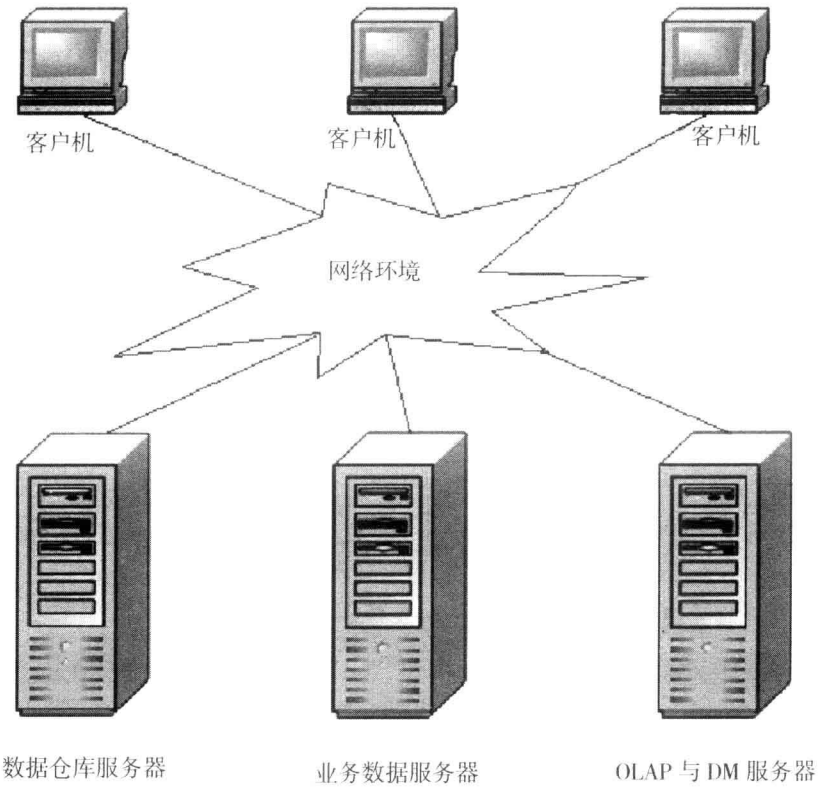


图 10 - 12　房地产决策支持系统部署图

（2）系统界面

决策支持系统登录页如图 10 - 13 所示。用户通过此登录界面可以进入 OLAP 报表系统和数据挖掘系统。

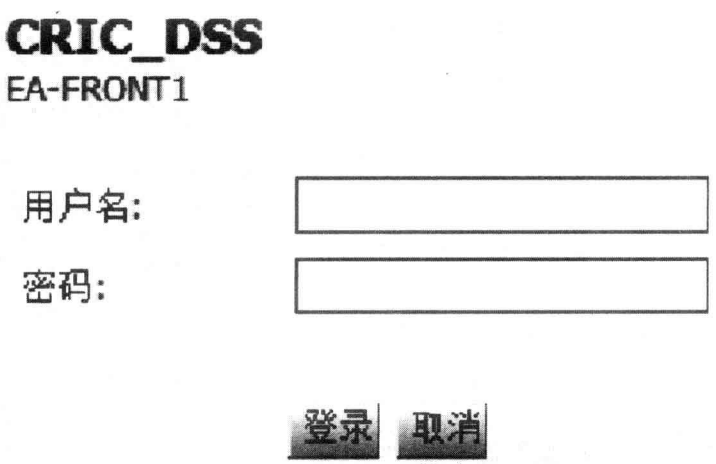


图 10 - 13　决策支持系统用户登录界面

OLAP 报表系统登录后主界面如图 10 - 14 所示。通过此系统可以查看各类报表（以域名 http://esf.sina.com.cn 为例）。

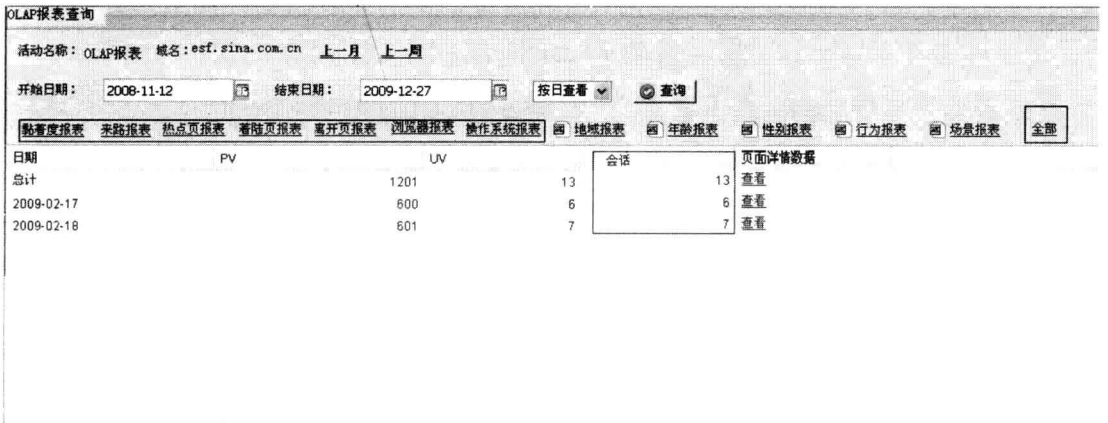
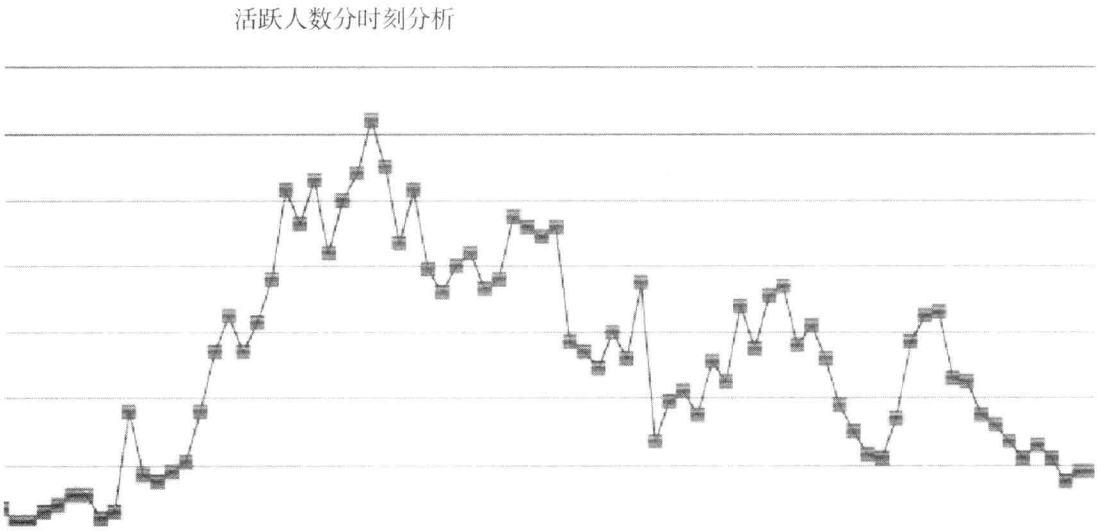


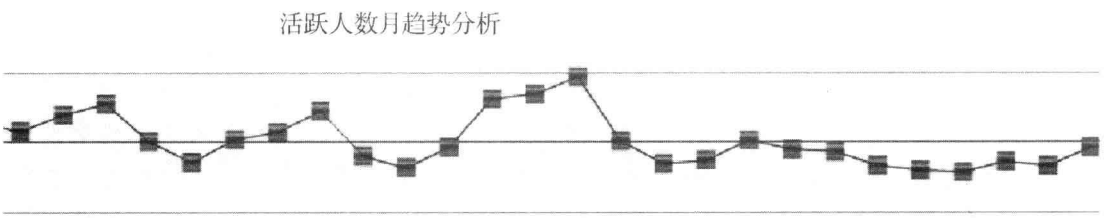
图 10 - 14　OLAP 报表系统主界面

（3）OLAP 系统功能实现

在本房地产决策支持系统中，所有与时间维、地理维相关的报表，都能在系统上直接实现时间维和地理维上的挖掘。如从经纪人每日套餐订购金额，对照时间维层系结构，可以直接得到套餐订购的周报、月报、季报与年报，通过导出度量直接获取周增长率、月收益增长率、季收益增长率及年度收益增长率等。根据经纪人所属的区域，对照地理维层系结构，可以得出区域套餐订购金额，从而直接获取城市套餐订购及省套餐订购情况。对于其他与时间、地理相关的报表，都能同样实现相对应的功能。示例：图 10 - 15 是关于时间层系上的挖掘。活跃人数分时刻分析是 15 分钟活跃人数，活跃人数月趋势分析是全天的活跃人数总和组成的日序列图。



（a）时间层系挖掘数据样例——活跃人数每日按时刻 15 分钟分析曲线



（b）时间层系挖掘数据样例——活跃人数每月趋势分析曲线

图 10－15

综合仪表盘示例（以网盟推广为例），如图 10－16 所示（图中数据为示例，非真实数据）。

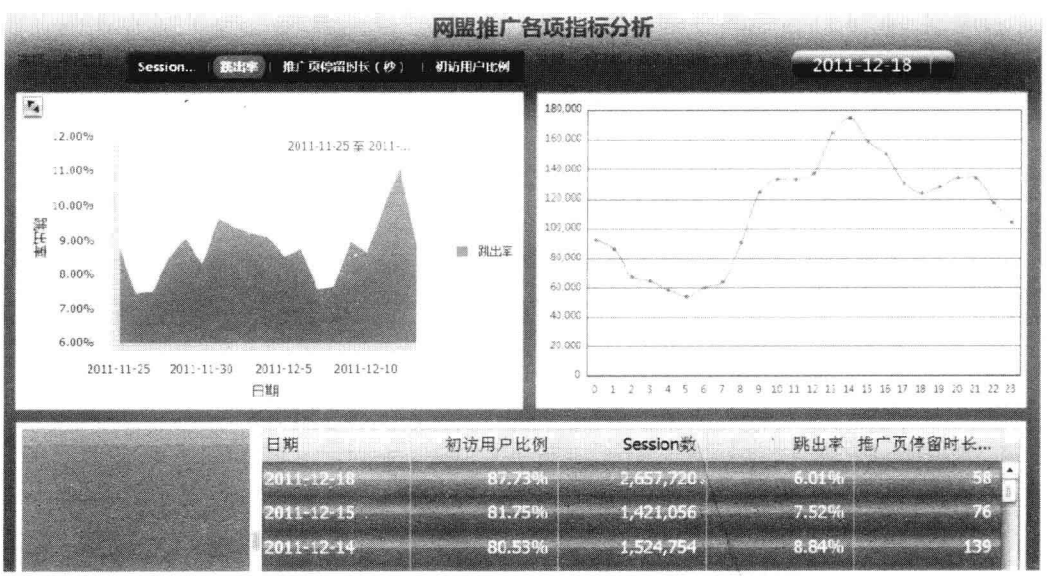


图 10－16　综合仪表盘示例

（4）楼市热点与用户行为结合实现精准推送房源

精准推送是现代网络营销的一个新概念，通过建立营销数据库信息来实现网络用户与商品的精准配对，而实现地点对点的营销方式。具体实现可以通过网络搜索结果、E－mail或是移动短信等方式来实现。根据本论文提出的数据仓库模型，我们可以实现房源的精准推送。

首先，我们可以在不同维度的各种层面上对楼市的热度进行分析。例如：可以查看一段时间或某个时间点的房源点击量和搜索量在物业类型这个层面上的走势。有了这些基于该 Web 数据仓库的 MOLAP 分析，决策者可以更快、更高效地获取有用信息、精准营销以及有针对性地分析各种因素对楼市的影响。

其次，我们也可以对房源点击量、刷新量和搜索量在时间维度、用户维度、房源维度和房源所属人维度上进行聚合分析。在时间维度上，可以分析不同层系下房源的点击量、刷新量和搜索量；在用户维度上我们可以检验访问用户性别可能对房源的喜好的倾向，以及职业、收入等属性对房源的偏好；在房源维度上，我们可以分析房源属性，例如面积，物业类型等，在一定时间内的受欢迎程度；在房源所属人维度上，可以按所属人的收入，教育程度等属性来分析所属人这些属性和其所提供房源的热度之间的关系。

再次，导航路径事实数据就是记录的用户访问网络的行为，根据对用户访问路径的分析，我们可以根据用户的普遍访问行为，找到用户访问的几类关键路径，同样我们可以对用户具体行为进行分类统计，如各用户在网站的停留时长，具体到各页面停留时长，用户访问的 PV 数。同时还可以通过对新注册用户的日增长率或是月增长率来判断网络运营的效果。通过对导航路径事实数据的 MOLAP 分析，网站设计者可以对更好地把握网站产品的设计，了解用户的实际需求。

最后将分析好的分类的热点房源信息，与按照网络行为的分类用户信息通过关联规则建立关联，来获取用户与房源的匹配关系，从而实现房源的精准推送。自从建立起房源精准推送机制，每日通过 EDM（Email Direct Marketing）平台向用户推送房源信息。这种营销策略从很大程度上减少了营销开销，而从用户角度来思考，让用户以最简易的方式来获取最有效的信息。

参考文献

[1] 李雄飞，李军. 知识发现与数据挖掘. 北京：高等教育出版社，2003：10.
[2] 徐洁磐. 数据仓库和决策支持系统. 北京：科学出版社，2005：45.
[3] 王晓东. 算法设计与分析. 北京：清华大学出版社，2003：5－10.
[4] 史忠植. 知识发现. 北京：清华大学出版社，2003：6－11.
[5] R. Groth 著. 数据挖掘构筑企业竞争优势. 侯迪译. 西安：西安交通大学出版社，2001：146－186.
[6] J. R. Quinlan. InductiongofgDecisionTree [M]. MachineLearning. 1986，1：230－330.
[7] J. R. Qulinlan. C4. 5：Programs for Machine Learning. Morgan Kaufmann Publisher，San Mateo，CA. 1993：27－48.
[8] Manish Mehta，Jorma Rissanen，Rakesh Agrawal. MDL－based decision tree pruning. International Conference on Knowledge Discovery in Databases and Data Mining. 1995：216－221.
[9] Rastogi R，Shim K. PUBLIC. A decision Tree Classifier that integrated Building and Pruning. In Proceeding 1998 International conference Very Large DataBase. NewYork. 1998：404－415.
[10] Manish Metha，Jirma Rissannen and Rakesh Agrawal. SLIQ：A Fast Sealable ClassifierforDatamining. InEDBT' 96，Avignon，France. 1996：227－248.
[11] Shafer J，Agrawal R，Mehta M. SPRINT：A Scalable Parallel Classifier for DataMining. In Proceeding 1996 Internatioanl conference Very Large DataBases. Bombay，India. 1996：544－555.
[12] 陈沛玲. 决策树分类算法优化研究. 中南大学论文，2007，25（6）：125－146.
[13] K. Alasabti，S. Ranka and V. singh. CLOUDS：A Decision Tree Classifier for Large Datasets [M]. 1998：102－134.
[14] Breiman L，Friedman J，Olshen R A. Classifieation and regression trees [Ml. Belmont：Wadsworth，1984：98－123.
[15] M. Mehta，R. Agrawal and J. Rissanen. SLIQ：A fast scalable classifier for datamining. In Proc of the Fifth Int' l Conference on Extending Database Technology，Avignon，France. 1996：14－48.
[16] 刘小虎，李生. 决策树的优化算法. 软件学报，1998，9（10）：797－800.
[17] 吴菲，黄梯云. 遗传算法在基于二元决策树的模型选择中的应用. 计算机研究与发展，1999，36（11）：1323－1328.
[18] 郭茂祖，刘扬. 一种新的基于属性—值对的决策树归纳算法 [J]. 小型微型计算机系统，2001，24（4）：459－462.
[19] 杨宏伟，赵明华，孙娟，等. 基于层次分解的决策树. 计算机工程与应用：2003，14（1）：108－110.
[20] 黄定轩，武振业. 一类加权连续属性的多变量决策树构造方法. 系统工程理论方法应用，2005，14（1）：80－83.
[21] 阳东升，彭小宏，修保新，等. 组织协作网与决策树. 系统工程与电子技术：2006，28（1）：63－67.

[22] Michalski, R. S, Carbonell, J. G. and Mltehell 著. 机器学习 - 实现人工智能的途径. 王树林译. 北京：科学出版社，1992：97 - 103.
[23] 梁循. 数据挖掘算法与应用. 北京：北京大学出版社，2006：45 - 48.
[24] Jiawei Han and Micheline Kamber. 数据挖掘概念与技术. 北京：机械工业出版社，2001：23 - 34.
[25] 万晓莉. 数据挖掘中的分类预测及其在决策支持系统中的应用研究. 成都：西南交通大学，2003：3 - 4.
[26] 蒋秀英. 数据挖掘技术的应用研究. 商场现代化，2008（10）：15 - 19.
[27] Swere E, Mulvaney D J. Robot Navigation Using Decision Trees. Loughborougk, UK: Elect - ronic Systems and Control Division Research, 2003.
[28] 毛国军. 数据挖掘原理与算法. 北京：清华大学出版社，2005：109 - 153.
[29] 石彦芳，石建国，周檬. 数据挖掘技术在 CRM 中的应用. 中国商贸，2010，02：73 - 74.
[30] 陈安，陈宁，周龙骧. 数据挖掘技术及应用. 北京：科学出版社，2006：41 - 47.
[31] 孙瑾，许青松，陈燕燕. 基于遗传算法和支持向量机的银行个人信用评估. 统计与决策，2008，12：230 - 278.
[32] 王琪. 浅谈基于信用评估的个人客户综合授信. 经济研究导刊，2010（13）：62 - 63.
[33] 闪四清，陈茵，程雁，等. 数据挖掘—概念、模型、方法和算法. 北京：清华大学出版社，2003：20 - 52.
[34] Jiawei Han, Micheline Kamber 著. 数据挖掘概念与技术. 范明，孟小峰译. 北京：机械工业出版社，2008：130 - 168.
[35] 杨静，张楠男，李建，等. 决策树算法的研究与应用. 计算机技术与发展，2010（02）：114 - 116，120.
[36] 蔡皎洁，张玉峰. 基于数据挖掘银行客户信用风险评级体系研究. 情报杂志，2010（02）：47 - 50，71.
[37] 王海. 数据仓库和数据挖掘在信用社客户关系管理中的应用. 计算机与现代化，2010（01）：162 - 165.
[38] 黄文. 决策树的经典算法：ID3 与 C4.5. 四川文理学院学报（自然科学版），2007，17（5）：16 - 18.
[39] 冯少荣. 决策树算法的研究与改进. 厦门大学学报（自然科学版），2007，46（4）：496 - 500.
[40] Agrawal R, Lmielinshi T, SwmiA. Database Mining: A Performance Perspective. IEEE Trans. on Knowledge and Data Engineering. 1993, 5（6）: 914 - 925.
[41] T. Fukuda, Y. Moroto and S. Morishita. Constructing efficient decision trees by using optimized numeric association rules. VLDB1996.
[42] Davies S, Moore A. Bayesian Networks for Lossless Dataset Compression. In Proceeding 1999 Intemational conference Knowledge Discovery and DataMining（KDD'99）. SanDiego, CA. Aug. 1999: 387 - 391.
[43] 曲开社，成文丽，王俊红. ID3 算法的一种改进算法. 计算机工程与应用，2003，9（25）：104 - 107.
[44] 卫金茂，王石. 基于 XML 的数据挖掘，计算机工程与设计，2003，24（10）.
[45] 徐勇，成良玉，李猛. 基于 XML 数据的关联规则挖掘研. 计算机工程与设计，2006，27（24）：4704 - 4706.
[46] 刘小虎，李生. 决策树的优化算法. 软件学报. 1998：6 - 160.
[47] 黄梯云. 智能决策支持系统. 北京：电子工业出版社，2001.
[48] 王万森. 人工智能原理及应用. 北京：电子工业出版社，2000.

[49] 史忠植. 知识发现. 北京：清华大学出版社，2002.
[50] 刘小群. 基于数据挖掘房地产决策支持系统的研究和开发. 上海交通大学工程硕士学位论文，2012：14－30，40－42.
[51] 林榕. 基于 O LAP 的学生就业指导数据分析. Equipment Manufactfing Technology，2007（8）.
[52] 薛薇、陈欢歌. Clementine 数据挖掘方法及应用. 北京：电子工业出版社，2010，9.
[53] C Kruegel，G Vigna，Anomaly detection of web－based attacks，in Proceedings of the 10th ACM conference on Computer and Communications security，pp. 251－256，2003.
[54] Jian Pei，Jiawei Han，Behzad Mortazavi－asl，Hua Zhu，Mining Access Patterns Efficiently from Web Logs，Knowledge Discovery and Data Mining Current Issues and New Applications，Volume 1805/2000，pp. 396－407，2000.
[55] Raymond McLeod，Jr. and George Schell，Management Information Systems Tenth Edition，published by Pearson Education，Inc.，Publishing as Prentice Hall.
[56] 杨喜孙，薛瑞鑫，叶华. 农村信用社小额信贷可持续发展面临的问题及对策. 农村经济. 2007，（5）：57－61.
[57] 李正波，高杰，崔卫东. 农村信用社农户贷款的信用风台县评价研究. 北京电子科技学院学报，2006，14（1）：70－74.
[58] 朱慧明. 现代贝叶斯统计理论的基本观点与研究现状. 江苏统计，2003，01：12－13.
[59] 李旭升，郭耀煌. 一种新颖混合贝叶斯分类模型研究. 计算机科学，2006，33（9）.
[60] 王久铱. 基于聚类的税务稽查选案方法及其系统的研究. 大连理工大学，2007.
[61] 赵斌. 关联规则分布式挖掘算法研究和实现. 南京师范大学硕士学位论文，2003.
[62] Mia K. Stern，Joseph E. Beverly Park Woof. Native Bayes Classifiers for User Modeling 2006.
[63] 张剑飞. 贝叶斯网络学习方法和算法研究. 东北师范大学硕士学位论文，2005.
[64] 蔡春华. 数据挖掘在高校学生成绩分析中的应用与研究. 哈尔滨理工大学工程硕士学位论文，2010.
[65] 谈恒贵，王文杰. 数据挖掘中的聚类算法. 微型机与应用，2005（1）.
[66] 蒋宗礼. 人工神经网络导论. 北京：高等教育出版社出版，2001.
[67] 魏海坤. 神经网络结构设计的理论与方法. 北京：国防工业出版社，2005.
[68] 史国庆，赵庆生. 一种神经网络模型算法控制的研究. 太原工业大学学报，1994，25（3）：41－46.
[69] 谢季坚，刘承平. 模糊数学方法及其应用（第二版）. 武汉：华中理工大学出版社，2000（5）.
[70] 陈善广，鲍勇. BP 神经网络学习算法的研究. 应用基础与工程科学学报，1995，3：437－442.
[71] 朴金姬. 神经网络算法在风电场功率预测中的应用研究. 华北电力大学硕士学位论文，2011，3.
[72] 田雨波. 混合神经网络技术. 北京：科学出版社，2009：60－73.
[73] 马云潜，张学工. 支持向量机函数拟合在分形插值中的应用. 清华大学学报（自然科学版），2000，40（3）：76－78.
[74] GiorgioV. Gene expression data analysis of human lymphoma using support vector machines and output coding ensembles. Artificial Intelligence in Medicine，2002，26：281－304.
[75] Xut，Huorm，Wangp，et al. Power system transient stability assessment based on statistical learning theory. Proceedings of The Chinese Society for Electric Engineering，2003，3（11）：51－55.
[76] 马笑潇，黄席樾. 基于支持向量机的故障过程趋势预测研究. 系统仿真学报，2002，14（11）：1948－1951.

[77] I. J. Cox, M. Miller, T. P. Minka, P. Yianilos. An Optimized Interaction Strategy for Bayesian Relevance Feedback. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, 1998: 553 - 558.
[78] 薛毅. 支持向量机与数学规划. 北京：北京工业大学博士论文，2003.
[79] 崔万照，等. 基于模糊模型支持向量机的混沌时间序列预测. 物理学报，2005，54（7）：3009 - 3018.
[80] 张诏，张素，章琛曦，陈亚珠. 基于支持向量机的概率密度估计方法. 系统仿真学报，2005（17）：2355 - 2357.
[81] 陈伟. 关联规则挖掘算法的改进及其应用. 江苏科技大学硕士学位论文，2009.
[82] 颜雪松，蔡之华. 一种基于 Apriori 的高效关联规则挖掘算法的研究. 计算机工程与应用，2002，38（10）：209 - 211.
[83] Cheung D W. Maintenance of discovered association rules in large databases: an incremental updating technique. Proceedings of the 12th Internet Engineering, 1997, 9 (5): 813.
[84] 芦洁，刘志镜. 挖掘关联规则中对 Apriori 算法的一种改进. 微电子学与计算机，2006，23（9）：80 - 90.
[85] 王晓峰，王天然，赵越. 一种自顶向下挖掘长频繁项的有效方法. 计算机研究与发展，2004，41（1）：148 - 155.
[86] 朱玉全，孙志挥，季小俊. 基于频繁模式树的关联规则增量更新算法. 计算机学报，2003，26（1）：91 - 96.
[87] 方伍元. 关联规则挖掘中若干关键技术的研究及其在专利信息挖掘中的应用研究. 江苏科技大学硕士学位论文，2007：13 - 14.
[88] Joog Soo Park, Ming - Syan Chen, Philip S. Yu. An Effective Hash Based Algorithm for Mining Association Rules. Califormia. 1995: 175 - 186.
[89] 秦吉胜，宋瀚涛. 关联规则挖掘 AprioriHybrid 算法的研究和改进. 计算机工程，2004，30（17）：7 - 8.
[90] 吴常辉. 基于关联规则的数据挖掘方法及其在电子商务网站中的应用研究. 合肥工业大学硕士学位论文，2010.
[91] 陈文伟. 智能决策技术. 北京：电子工业出版，1998：5 - 6，114 - 149.
[92] 阎平凡，张长水. 人工神经网络与模拟进化计算. 北京：清华大学出版社，2000.
[93] Pawlak Z Rough Sets: Thoretical Aspects of Reasoning About Data. Kluwer Academic Publisher, 1991: 1 - 23.
[94] 郭雨松. 一种启发式贝叶斯分类算法及其在铁路货运客户细分中的应用研究. 北京交通大学硕士学位论文，2008，6.
[95] Moulin L S, Alvesap, Sharkawi Ma. Support vector machines for transient ability analysis of large-scale Power systems. IEEE Trans. On Power System. 2004, 19 (2): 818 - 825.
[96] 张磊，林福宗，张钹. 基于支持向量机的相关反馈图像检索算法. 清华大学学报（自然科学版），2002，42（1）：80 - 83.
[97] 刘海涛. 支持向量机分类算法的研究. 江南大学硕士学位论文，2009，5.
[98] 胡莹. 支持向量机的证券投资风险管理研究. 西安电子科技大学硕士学位论文，2010，1.
[99] 祝金荣. 基于支持向量机的石油期货价格预测. 工业技术经济，2007，26（2）：59 - 61.
[100] 卢硕. 数据仓库和数据挖掘在决策支持系统中的应用研究. 西安电子科技大学硕士学位论文，2006（1）.
[101] 刘明德，陈湘，林杰斌. 数据挖掘与 OLAP：理论与实务. 北京：清华大学出版社，2003.
[102] Inmon W H 著. 数据仓库. 王志海译. 北京：机械工业出版社，2000，5 - 60.

[103] 孙栋，陈莉，宋亚奇. 基于数据仓库的决策支持系统. 微机发展，2005，15（10）：56－59
[104] 左金柱. 基于数据仓库的商业决策支持系统的研究与应用. 北京科技大学，2004.
[105] 潘华，项同德. 数据仓库与数据挖掘原理、工具及应用. 北京：中国电力出版社，2007：9－5，105－112.
[106] 张亦军. 基于粗糙集和遗传算法的大数据集数据挖掘应用研究. 太原理工大学，2007.
[107] 夏春艳. 基于粗集属性约简的数据挖掘技术的研究与应用. 长春理工大学硕士学位论文，2007.
[108] 孔祥明. 基于变精度粗糙集的连续属性离散化方法及数据预处理方法. 东北师范大学，2006.
[109] 梁雪峰，等. 基于粗集理论和云理论的 GIS 数据挖掘与知识发现. 河北建筑科技学院学报，2004，21（2）.
[110] Mrozek A et al. Methodology of rough controller synthesis. Proc. of IEEEInternational Conference on Fuzzy Systems，New Jersey，1996，1135－1139.
[111] Slowinski R et al. Rough set sorting of firms according to bankruptcy risk. In：Applying Multiple Criteria aid for Decision to Environment Management，Dordrecht：Kluwer Academic Publishers，1994，339－357.
[112] 刘文军. 基于粗糙集的数据挖掘算法研究. 北京师范大学博士论文，2004，4.
[113] 曾庆茂. 基于神经网络和模糊推理的信息融合技术. 西安科技大学硕士学位论文，2005.
[114] 朱欢萍，朱清根，黎莉. 模糊多属性决策法在配送中心选址中的应用. 商业研究，2007（10）：28－32.
[115] 杨云，任建军，王建华，等. 模糊多属性决策在装备质量评价中的应用. 数学的实践与认识，2005，35（3）：34－38.
[116] 姜明辉，王欢，王雅林. 分类树在个人信用评估中的应用. 商业研究，2003（21）：86－88.
[117] 方先明，熊鹏. 基于 RBF 网络的商业银行信用风险控制研究. 金融论坛，2005（4）.
[118] 李萌，陈柳钦. 基于 BP 神经网络的商业银行信用风险识别实证分析. 南京社会科学，2007（1）：18－29
[119] Yeun Y S，Lee K H，Han S M，Yang Y S. Multiple neural networks coupledwith oblique decision tree：a case study on the configuration design of midshipstructure. Intelligent Information Systems，1997，11（8－10）：161－167.
[120] Zmijewski M E. Methodological Issues Related to the Estimation of FinancialDistress Prediction Models. Journal of Accounting Research，1954：59－86.
[121] 蔡丽艳. 基于数据挖掘的农户小额贷款信用评估模型研究. 哈尔滨理工大学工程硕士学位论文，2011，6.
[122] 王琪. 基于数据挖掘的商业银行个人信用风险评估体系研究. 北京交通大学专业硕士学位论文，2009，6.
[123] 张玉婷. 我国消费信贷个人信用风险评估研究. 武汉理工大学，2007：4.
[124] 王宗军. 智能决策支持系统的结构模型及研究趋势. 决策与决策支持系统，2007，7（2）：49－54.
[125] 陈文伟. 决策支持系统教程［M］. 北京：清华大学出版社，2004.
[126] Taniar，D，Rahayu，W，Lee，V et al. Exception rules in association rule mining. Applied Mathematics and Computation，2008，205（2）：735－750.
[127] 俞文斌，张忠能，谢康林. 基于数据仓库的决策支持系统框架研究. 上海交通大学学报，2000，34（6）：810－812.

[General Information]
书名=数据挖掘算法及其应用研究
作者=蔡丽艳著
页数=182
SS号=13457229
DX号=
出版日期=2013.02
出版社=电子科技大学出版社

封面
书名
版权
前言
目录
第1章　概论
1.1　数据挖掘概念及其产生的背景
1.2　数据挖掘起源
1.3　数据挖掘任务
1.4　数据挖掘过程
1.5　数据挖掘解决的典型问题
1.6　数据挖掘常用工具及开发商
第2章　决策树算法
2.1　分类原理
2.2　决策树算法概述
2.3　决策树相关概念及定义
2.4　决策树生成过程
2.5　决策树算法特点
2.6　决策树剪枝技术
2.7　决策树算法国内外研究现状
2.8　ID3算法
2.9　C4.5算法
2.10　CART算法
2.11　SLIQ算法
2.12　SPRINT算法
2.13　PUBLIC算法
2.14　决策树算法比较
第3章　贝叶斯分类算法
3.1　贝叶斯分类理论
3.2　贝叶斯分类模型
3.3　贝叶斯分类模型的改进研究
3.4　贝叶斯网络
3.5　贝叶斯网络应用实例
第4章　人工神经网络算法
4.1　基本概念及特征
4.2　人工神经网络的分类
4.3　人工神经网络的发展历史
4.4　人工神经元模型
4.5　BP神经网络
4.6　BP神经算法应用实例
第5章　支持向量机

5.1 概述
5.2 统计学习理论
5.3 支持向量机定义
5.4 核函数
5.5 支持向量机的训练算法
5.6 支持向量机应用实例
第6章 关联规则
6.1 关联规则相关定义及基本概念
6.2 APRIORI算法
6.3 从频繁项集到强关联规则
6.4 提高APRIORI算法的效率
6.5 Web挖掘
第7章 聚类分析
7.1 概述
7.2 数据结构与数据类型
7.3 聚类准则函数
7.4 主要聚类算法
7.5 K-means聚类算法
7.6 K-means算法应用实例
第8章 数据挖掘相关技术
8.1 粗糙集技术
8.2 模糊集技术
8.3 数据仓库
第9章 数据挖掘在信用评估模型中的应用
9.1 我国商业银行运用数据挖掘技术的必要性
9.2 信用的基本概念
9.3 信用风险的影响因素
9.4 风险评估的特点
9.5 信用评估模型的主要应用技术
9.6 基于数据挖掘的农户小额贷款信用评估模型研究
第10章 数据挖掘在决策支持系统中的应用
10.1 决策支持系统
10.2 决策支持系统的主要特征及功能
10.3 智能决策支持系统
10.4 基于数据挖掘的房地产决策支持系统的开发
参考文献